AN ABSTRACT OF THE THESIS OF

Gordon F. Norris for the degree of Master of Science in Mathematics presented on September 22, 1999. Title: Spectral Integration and the Numerical Solution of Two-Point Boundary Value Problems.

Abstract approved: _____
Satish Reddy

Spectral integration methods have been introduced for constant-coefficient two-point boundary value problems by Greengard, and pseudospectral integration methods for Volterra integral equations have been investigated by Kauthen. This thesis presents an approach to variable-coefficient two-point boundary value problems which employs pseudospectral integration methods to solve an equivalent integral equation.

This thesis covers three topics in the application of spectral integration methods to two-point boundary value problems.

The first topic is the development of the spectral integration concept and a derivation of the spectral integration matrices. The derivation utilizes the discrete Chebyshev transform and leads to a stable algorithm for generating the integration matrices. Convergence theory for spectral integration of C^k and analytic functions is presented. Matrix-free implementations are discussed with an emphasis on computational efficiency.

The second topic is the transformation of boundary value problems to equivalent Fredholm integral equations and discretization of the resulting integral equations. The discussion of boundary condition treatments includes Dirichlet, Neumann, and Robin type boundary conditions.

The final topic is a numerical comparison of the spectral integration and spectral differentiation approaches to two-point boundary value problems. Numerical results are presented on the accuracy and efficiency of these two methods applied to a set of model problems.

The main theoretical result of this thesis is a proof that the error in spectral integration of analytic functions decays exponentially with the number of discretization points N. It is demonstrated that spectrally accurate solutions to variable-coefficient boundary value problems can be obtained in $O(N\log N)$ operations by the spectral integration method. Numerical examples show that spectral integration methods are competitive with spectral differentiation methods in terms of accuracy and efficiency.

Spectral Integration and the Numerical Solution of Two-Point Boundary Value Problems

by

Gordon F. Norris

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 22, 1999
Commencement June 2000

Master of Science thesis of Gordon F. Norris presented on September 22, 1999

APPROVED:

Redacted for Privacy

Major Professor, representing Mathematics

Redacted for Privacy

Chair of Department of Mathematics

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Gordon F. Norris

TABLE OF CONTENTS

Page

1. Introduction 1

2. Spectral Integration 8

2.1 The Spectral Integration Matrices S_l and S_r 8
2.2 Spectral integration of analytic functions 12
2.3 Efficient computation of the spectral integration matrices 17
2.4 Matrix-free implementation with the FFT 23

3. Integral Formulation and Discretization 26

3.1 Integral formulation of boundary value problems 26
3.2 Discrete analogues of the operators K and J 28
3.3 The treatment of Neumann and Robin boundary conditions 30

4. The Solution of Boundary Value Problems: Numerical Examples 33

4.1 An elementary constant coefficient equation 33
4.2 A variable-coefficient problem 39
4.3 Nonlinear boundary value problems 43

5. Conclusions 48

BIBLIOGRAPHY 50

APPENDICES 51

Appendix 1 The CGS Method 52
Appendix 2 Spectral Differentiation Implementation Details 54
Appendix 3 Codes for problem (4.1) 57

LIST OF FIGURES

Figure		Page
2.1	Infinity norm of the spectral integration error	12
2.2	E_ρ	14
2.3	Work (flops): function INTMAT	22
2.4	Work (flops) to apply $\boldsymbol{S}_I$: matrix vs. transform based integration	25
4.1.1	Error in the solution of (4.1) with $\mu = 1$	34
4.1.2	Efficiency of spectral methods in the solution of (4.1)	35
4.1.3	Error in the iterative solutions of (4.1) for $\mu = 1$	36
4.1.4	Efficiency of iterative solutions to (4.1) with $\mu = 1$	37
4.1.5	Efficiency comparison: solution of (4.1) with $\mu = 100$	38
4.2.1	Error vs. work for spectral solutions of (4.2)	40
4.2.2	Convergence of the CGS iteration	41
4.2.3	Work/Nlog(N) for the spectral integration solution of (4.2)	42
4.3.1	Solution of equation (4.3)	44
4.3.2	Progress of the iteration for $N = 32$	45
4.3.3	Solution of equation (4.5)	46
4.3.4	Progress of the iteration for $N = 32$	46

LIST OF TABLES

Table Page

4.1 Condition numbers.....34

4.2 Number of iterations.....37

4.3 Number of iterations.....39

4.4 Execution times.....42

Spectral Integration and the Numerical Solution of Two-Point Boundary Value Problems

Chapter 1
Introduction

This thesis investigates the spectral integration method for solving differential boundary value problems, and compares it to the spectral differentiation method. Over the past three decades, highly accurate numerical methods for the solution of differential equations have been developed based on spectral differentiation. The distinguishing feature of spectral methods is their so-called *spectral accuracy*. For functions analytic on the interval of interest, the error in spectral differentiation decays exponentially with the number of discretization points. This is in contrast with finite difference or finite element methods which converge algebraically. For example, in second order finite difference approximations the error typically decays like N^{-2} as the number of discretization points N increases.

Two distinct types of spectral methods have emerged, Galerkin/Tau and collocation (or pseudospectral). In the Galerkin/Tau formulation a solution is sought in the form of a truncated series expansion of orthogonal basis functions. Typical bases for approximation on bounded intervals are trigonometric functions and Chebyshev or Legendre polynomials. Differentiation in Galerkin/Tau methods is done by exactly differentiating the series term by term. In problems with variable coefficients the complication of determining the expansion coefficients of products arises, i.e. the evaluation of convolution sums. The principal advantage of the Galerkin/Tau procedure for numerical computation is that it results in sparse linear systems for the expansion coefficients, which may be solved efficiently via banded Gaussian elimination. In the presence of general convolution sums, one loses this advantage, as dense linear systems

replace sparse ones. As a consequence, applications are usually limited to linear constant-coefficient equations.

In spectral collocation methods, functions are represented by the values they assume on a finite set of nodes, or collocation points. This is equivalent to representation by the coefficients of an expansion in Lagrange interpolation polynomials. Let $p_N(x)$ be a polynomial of degree N satisfying the boundary conditions. Imposing the condition that $p_N(x)$ satisfy the differential equation at each of the interior interpolation points leads to a system of linear algebraic equations. The solution of this system of equations gives the approximate solution of the boundary value problem.

In this paper we focus on Chebyshev pseudospectral methods which utilize the Chebyshev extreme points as interpolation nodes. These Chebyshev extreme points, hereafter referred to as simply the *Chebyshev points*, are the extrema of $T_N(x)$ $x \in [-1,1]$ denoted by

$$\{x_j\} \qquad x_j = \cos(j\pi/N) \qquad j = 0,1,...,N.$$

A well known result from approximation theory [10] is that polynomial interpolation at the Chebyshev points is nearly optimal in the sense of minimizing the maximum interpolation error.

Chebyshev pseudospectral differentiation consists of interpolating a function at the Chebyshev points with a polynomial and approximating the derivative of the function at the Chebyshev points by the derivative of the interpolating polynomial. In exact arithmetic this process is exact for polynomials up to degree N. Spectral differentiation is usually implemented as a matrix- vector product

$$\boldsymbol{f}^{(1)} = \boldsymbol{D}_N \boldsymbol{f},$$

where $\boldsymbol{f}$ is a vector of function values at the nodes, $f_j = f(x_j)$, $\boldsymbol{f}^{(1)}$ is a vector of approximate derivative values $f_j^{(1)} \approx \frac{df}{dx}(x_j)$, and $\boldsymbol{D}_N$ is the spectral differentiation matrix. The matrix-vector product $\boldsymbol{D}_N\boldsymbol{f}$ can be performed in $N\log N$ operations by a procedure that uses the discrete Fourier transform. See [3],[4] for details on differentiation matrices and transform based differentiation.

The pseudospectral approach has proven to be a powerful tool for the solution of boundary value problems. The straightforward treatment of variable coefficients and nonlinearities, combined with exceptional accuracy, has contributed to the increasing popularity of these methods. For details on the theory and implementation of spectral methods we refer to [3],[4],[5].

Spectral differentiation, however, is not without its drawbacks. The differentiation matrices become ill-conditioned with increasing N, the degree of the interpolating polynomial. To illustrate the significance of this point, consider solving the following model problem using spectral differentiation methods.

$$u'' - u = f(x), \qquad u(-1) = u(1) = 0. \tag{1.1}$$

The spectral differentiation approach to this problem results in the linear system

$$(\boldsymbol{D}_N^{(2)} - \boldsymbol{I})\boldsymbol{u} = \boldsymbol{f}$$

for $\boldsymbol{u}$, where u_j is the approximation to $u(x_j)$, $\boldsymbol{D}_N^{(2)}$ is the second order spectral differentiation matrix incorporating homogeneous Dirichlet boundary conditions, and $\boldsymbol{I}$ is the identity matrix. The spectral radius of $\boldsymbol{D}_N^{(2)}$ is known to be $O(N^4)$, and the magnitude of the smallest eigenvalue is approximately $\frac{\pi^2}{4}$ [5]. $\boldsymbol{D}_N^{(2)}$ is a non-normal matrix so the spectral radius does not equal the 2-norm, however the spectral radius does give us a lower bound for the 2-norm. It then follows that the 2-norm condition number of the

collocation matrix, $\kappa_2(\boldsymbol{D}_N^{(2)} - \boldsymbol{I})$ is at least $O(N^4)$ as $N \to \infty$. Numerical computations strongly suggest that $\kappa_2(\boldsymbol{D}_N^{(2)} - \boldsymbol{I}) > C(N^4)$ for all N, where C is a constant. In floating point arithmetic computations, this condition number imposes an upper limit on the order of the approximation. Let ϵ_m denote machine epsilon ($\epsilon_m \approx 10^{-16}$ in IEEE double precision). Clearly, if $\kappa_2(\boldsymbol{D}_N^{(2)} - \boldsymbol{I}) \approx \epsilon_m^{-1}$ the method essentially fails; rounding errors in the data result in $O(1)$ errors in the solution [6].

Next we consider the integral collocation approach to the model problem (1.1), but first a brief description of spectral integration. Given a function f defined on [-1,1], interpolate f at the Chebyshev points by a polynomial p of degree N. Approximate $\int_{-1}^{x_j} f(s)ds$ by $\int_{-1}^{x_j} p(s)ds$, where the integral of p is computed exactly. Spectral integration is also implemented as a matrix-vector product, and is discussed in detail in Chapter 2.

Let G denote the Green's function for the one-dimensional Poisson equation with homogeneous Dirichlet conditions $u'' = f(x), \qquad u(\pm 1) = 0.$

$$G(x,s) = \begin{cases} (x-1)(s+1)/2 & -1 \le s \le x \\ (x+1)(s-1)/2 & x \le s \le 1 \end{cases}$$

Using this Green's function, it can be shown that the boundary value problem (1.1) can be reformulated as a Fredholm integral equation

$$u(x) - \int_{-1}^{1} G(x,s)u(s)ds = \int_{-1}^{1} G(x,s)f(s)ds.$$

In operator notation: $u - Ku = Kf$. In the integral collocation approach, we approximate this integral equation by a system of linear algebraic equations

$$\boldsymbol{u} - \boldsymbol{K}_N\boldsymbol{u} = \boldsymbol{K}_N\boldsymbol{f} \quad \text{or} \quad (\boldsymbol{I} - \boldsymbol{K}_N)\boldsymbol{u} = \boldsymbol{K}_N\boldsymbol{f},$$

where $\boldsymbol{u}$ and $\boldsymbol{f}$ are as described above in the differential collocation approach and $\boldsymbol{K}_N$ is a discrete analogue of the operator K. (A precise definition of $\boldsymbol{K}_N$ will be given in Chapter 3). Numerical evidence strongly suggests that $\kappa_2(\boldsymbol{I} - \boldsymbol{K}_N)$ is uniformly bounded in N. We

conjecture that $\kappa_2(\boldsymbol{I} - \boldsymbol{K}_N) < 1.5$ for all N. Thus the method is numerically stable; small perturbations in the data correspond to small perturbations of the solution. In addition we note that in the integral formulation matrix conditioning does not impose an upper limit on the order of the approximation.

As this example shows, the spectral integration methods which we introduce are not subject to the inherent ill-conditioning of spectral differentiation. Other authors have considered spectral integration. Greengard [7] proposed an efficient Galerkin scheme for linear constant-coefficient boundary value problems. One of the notable features of Greengard's algorithm is that it achieves spectral accuracy in $O(N\log N)$ flops (floating point operations). A distinct disadvantage is that the method is only applicable to constant-coefficient equations.

Greengard's approach to boundary value problems of the form

$$u'' + \mu u' + \nu u = f(x), \quad u(-1) = \alpha, \quad u(1) = \beta$$

is to convert the problem to an integral equation for u''. The functions u'' and f are represented by truncated Chebyshev series, and the resulting discrete equations are pentadiagonal except for two rows arising from the boundary conditions. This discrete problem is solved directly in approximately $10N$ floating point operations. The algorithm requires two cosine transforms in addition to the solution of the linear system for a total estimated operation count of $10N(\log N - 1)$ flops.

Kauthen [8] presented a pseudospectral method for Volterra equations. Kauthen's philosophy is similar to our own, but he employs an ill-conditioned method for computing the spectral integration matrix.

The methods we propose are applicable to variable coefficient problems, and are not plagued by the conditioning problems of Kauthen's method. In our approach it takes $O(N^2 \log N)$ operations to create the matrices which represent the discretized integral equation, however if one only needs to compute matrix-vector products this can be done in $O(N \log N)$ operations. Jean-Paul Kauthen and my advisor, Dr. Satish Reddy suggested the basic idea behind this work: pseudospectral integration employing Chebyshev polynomials as basis functions should lead to well-conditioned collocation methods.

Our motivation for studying spectral integration techniques is based on a fundamental difference between differential and integral operators. Differential operators are unbounded (in appropriate norms), and this property is reflected in their discrete approximations, such as the spectral differentiation matrices. The integral operators on bounded domains which we consider are bounded linear operators and we expect to see this property manifested in their discrete analogues.

This thesis focuses on the following fundamental questions:

(1) Can spectrally accurate solutions to variable-coefficient boundary value problems be obtained in $N \log N$ operations using the spectral integration method?

(2) How does the efficiency of the spectral integration method compare with that for the spectral differentiation approach?

The answer to question (1) is yes. This is accomplished by using a fast algorithm to compute matrix-vector products and an iterative method to solve the resulting linear systems. Measuring the work to achieve a given accuracy we find the efficiency of the

two approaches is comparable, however one must employ a preconditioned iterative method to solve the linear systems for the spectral differentiation approach.

The remainder of this thesis is organized as follows. In Chapter 2 we define the spectral integration matrices and present a theorem on the convergence of spectral integration for analytic functions. In addition, we propose an efficient algorithm for generating the integration matrices, and discuss a matrix-free fast transform implementation. Chapter 3 covers the transformation of linear boundary value problems to integral equations, the treatment of boundary conditions, and spectral discretization. In Chapter 4 we present numerical examples of the solution of boundary value problems via spectral integration. The final chapter consists of a summary of our results, and suggestions for related research topics.

Chapter 2
Spectral Integration

2.1 The Spectral Integration Matrices S_l and S_r

This section is devoted to a detailed description of spectral integration, which leads us to a formal definition of the integration matrices.

Let $p_N(x)$ be the polynomial of degree N which interpolates $f(x)$ at the Chebyshev points $p_N(x_j) = f(x_j)$. We define spectral integration on the Chebyshev points as either of the two approximations

$$\int_{-1}^{x_j} f(s)ds \approx \int_{-1}^{x_j} p_N(s)ds, \qquad \int_{x_j}^{1} f(s)ds \approx \int_{x_j}^{1} p_N(s)ds.$$

$p_N(x)$ may be expressed as a linear combination of Lagrange interpolation polynomials

$$p_N(x) = \sum_{j=0}^{N} f_j L_j(x), \qquad L_j(x) = \prod_{\substack{k=0 \\ k\neq j}}^{N} \frac{x - x_k}{x_j - x_k}.$$

Integration of $L_j(x)$ requires repeated integration by parts and yields a result which is not expressible in terms of $\{L_j(x)\}_{j=0}^{N}$, thus we seek an alternate representation. An equivalent representation of p_N is obtained by taking the Chebyshev polynomials $\{T_j(x)\}_{j=0}^{N}$ as a basis for $\boldsymbol{P}_N$. In the Chebyshev basis $p_N(x) = \sum_{j=0}^{N} \hat{f}_j T_j(x)$, where $\hat{f} = (\hat{f}_0, \ldots, \hat{f}_N)^T$ denotes the discrete Chebyshev transform of $f(x)$.

With $f = (f_0, \ldots, f_N)^T = (f(x_0), \ldots f(x_N))^T$ the discrete Chebyshev transform and inverse transform are expressed concisely in matrix notation as

$$\hat{f} = Cf \qquad \text{and} \qquad f = C^{-1}\hat{f}$$

where $C_{jk} = \frac{2}{Nc_jc_k}\cos(jk\pi/N)$, $\quad (C^{-1})_{jk} = \cos(jk\pi/N)$ for $j,k = 0,1,2,\ldots,N$,

and $c_i = \begin{cases} 2 & i = 0, N \\ 1 & 1 \le i \le N-1 \end{cases}$ [3].

To develop a matrix representation of spectral integration on the Chebyshev points it is expedient to first define a transform space antidifferentiation matrix $\hat{A}$. Let $p(x) = \sum_{j=0}^{N} a_j T_j(x)$ and $q(x) = \sum_{j=0}^{N} b_j T_j(x)$, where $q(x)$ interpolates $p^{(-1)}(x)$ at the Chebyshev points. Here $p^{(-1)}$ denotes an antiderivative of p. Let $\boldsymbol{a} = (a_0, \ldots, a_N)^T$, $\boldsymbol{b} = (b_0, \ldots, b_N)^T$, and define $\hat{A}$ by the mapping $\hat{A}\boldsymbol{a} = \boldsymbol{b}$.

The Chebyshev polynomials of the first kind may be defined by [10]:

$T_0(x) = 1, \quad T_1(x) = x, \quad$ and $T_{j+1} = 2xT_j - T_{j-1}$ for $j \ge 1$.

We start with the first derivative recursion: $2T_j = \frac{T'_{j+1}}{j+1} - \frac{T'_{j-1}}{j-1}$ for $j \ge 2$. Antidifferentiating we obtain $T_j^{(-1)} = \frac{1}{2}\left(\frac{T_{j+1}}{j+1} - \frac{T_{j-1}}{j-1}\right)$ for $j \ge 2$. For $j = 1,2$ we take $T_0^{(-1)} = T_1$ and $T_1^{(-1)} = \frac{T_0 + T_2}{4}$. Note that the particular integration constants are irrelevant, as we are eventually going to use these antiderivatives to evaluate definite integrals.

We proceed by antidifferentiating the expansion of $p(x)$.

$$\sum_{j=0}^{N} a_j T_j^{(-1)} = a_0 T_1 + a_1 \frac{T_0 + T_2}{4} + \frac{1}{2} \sum_{j=2}^{N} a_j \left(\frac{T_{j+1}}{j+1} - \frac{T_{j-1}}{j-1}\right)$$

$$= \frac{a_1}{4} T_0 + a_0 T_1 + \frac{a_1}{4} T_2 + \sum_{j=3}^{N+1} a_{j-1} \frac{T_j}{2j} - \sum_{j=1}^{N-1} a_{j+1} \frac{T_j}{2j}$$

$$= \frac{a_1}{4} T_0 + (a_0 - \frac{a_2}{2}) T_1 + \sum_{j=2}^{N-1} \frac{a_{j-1} - a_{j+1}}{2j} T_j + \frac{a_{N-1}}{2N} T_N + \frac{a_N}{2(N+1)} T_{N+1}.$$

Notice that T_{N+1} and T_{N-1} alias one another on the Chebyshev points, i.e. T_{N-1} interpolates T_{N+1} at the Chebyshev points:

(2.1) $T_{N\pm1}(x_j) = \cos((N\pm 1)j\pi/N) = \cos(j\pi)\cos(j\pi/N) \pm \sin(j\pi)\sin(j\pi/N) = (-1)^j x_j.$

As the T_{N+1} term is to be evaluated at the Chebyshev points, it may be replaced by its alias without error. Thus, we set

$$b_{N-1} = \frac{a_{N-2} - a_N}{2(N-1)} + \frac{a_N}{2(N+1)} = \frac{a_{N-2}}{2(N-1)} - \frac{a_N}{N^2-1}.$$

In summary;

$$b_0 = \frac{a_1}{4}, \quad b_1 = a_0 - \frac{a_2}{2},$$

$$b_j = \frac{a_{j-1} - a_{j+1}}{2j} \quad \text{for} \quad j = 2,3,...,N-2,$$

$$b_{N-1} = \frac{a_{N-2}}{2(N-1)} - \frac{a_N}{N^2-1}, \quad \text{and} \quad b_N = \frac{a_{N-1}}{2N}.$$

Thus $\hat{A}$ is the tridiagonal matrix with non-zero entries

$$\hat{A}_{01} = \frac{1}{4}, \quad \hat{A}_{10} = 1$$

$$\hat{A}_{i,i-1} = \frac{1}{2i} \quad i = 2,3,...,N \quad \text{and} \quad \hat{A}_{i,i+1} = \frac{-1}{2i} \quad i = 1,2,...,N-2$$

$$\hat{A}_{N-1,N} = \frac{-1}{N^2-1}.$$

To put the role of $\hat{A}$ in perspective, let $\boldsymbol{p} = (p(x_0),p(x_1),...,p(x_N))^T$ and $\boldsymbol{q} = (q_0,...,q_N)^T = (q(x_0),...,q(x_N))^T$, where the q_j satisfy

(2.2) $$\int_{-1}^{x_j} p(s)ds = q_j - q_N \quad \text{and} \quad \int_{x_j}^{1} p(s)ds = q_0 - q_j \qquad j = 0,1,...,N.$$

Then

$$\boldsymbol{Cp} = \boldsymbol{a}, \qquad \widehat{\boldsymbol{A}}\boldsymbol{a} = \boldsymbol{b}, \text{ and } \qquad \boldsymbol{C}^{-1}\boldsymbol{b} = \boldsymbol{q}.$$

That is

$$\boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}\boldsymbol{C}\boldsymbol{p} = \boldsymbol{q}.$$

Thus we define the (physical space) antidifferentiation matrix $\boldsymbol{A}$ related to $\widehat{\boldsymbol{A}}$ by the similarity transformation $\boldsymbol{A} = \boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}\boldsymbol{C}$. The final step in the derivation is to express the differences in (2.2) in terms of on operation on $\boldsymbol{q}$. Consider the projector $\boldsymbol{P}$, whose action on a vector is to subtract the last element from each element of the vector

$$\boldsymbol{P}\boldsymbol{q} = (q_0 - q_N, q_1 - q_N, \ldots, q_N - q_N)^T,$$
and
$$-\boldsymbol{P}^C\boldsymbol{q} = (q_0 - q_0, q_0 - q_1, \ldots, q_0 - q_N)^T$$

where $\boldsymbol{P}^C$ denotes the centro-transpose $(P^C)_{ij} = P_{N-i,N-j}$. The entries of the matrix $\boldsymbol{P}$ are

$P_{ii} = 1$, and $P_{iN} = -1$ for $i = 0,1,\ldots,N-1$ with all other $P_{ij} = 0$.

We now define the left and right spectral integration matrices

(2.3) $$\boldsymbol{S}_l = \boldsymbol{P}\boldsymbol{A} = \boldsymbol{P}\boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}\boldsymbol{C} \quad \text{and} \quad \boldsymbol{S}_r = -\boldsymbol{P}^C\boldsymbol{A} = -\boldsymbol{P}^C\boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}\boldsymbol{C}.$$

Observe that generating the integration matrices directly from the definitions is *not* computationaly efficient, requiring $O(N^3)$ flops. An efficient algorithm for computation of the integration matrices is presented in a later section.

We conclude this section with a numerical example demonstrating the accuracy and convergence rate of spectral integration. We consider three functions; a $C^2[-1,1]$ function $f_1(x) = |x|^3$, a $C^\infty[-1,1]$ function $f_2(x) = (1+x^2)^{-1}$, and an entire function $f_3(x) = e^x$. Note that $f_2(z)$ is analytic save for simple poles at $z = \pm i$. Spectral integration was performed on these three functions with several different values of the parameter N.

The computed approximations are compared with the result of performing the integration analytically. The measure used is

$$error(N) = \max_{0 \le j \le N} \left| \int_{-1}^{x_j} f(x)dx - (\boldsymbol{S}_I \boldsymbol{f})_j \right|.$$

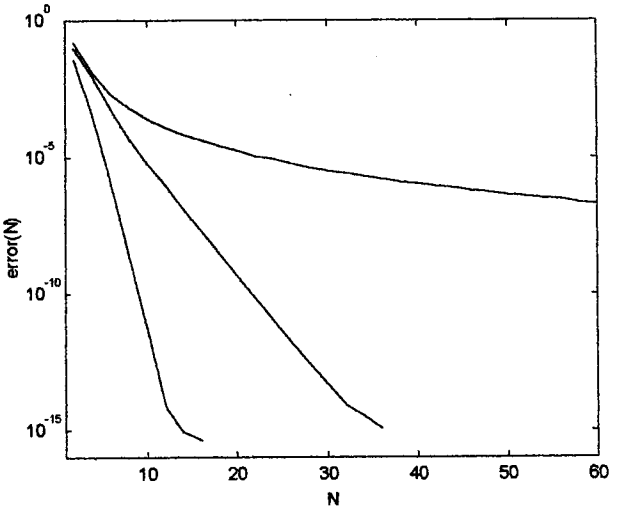

Figure 2.1: Infinity norm of the spectral integration error. The upper, middle,and lower curves correspond to f_1,f_2, and f_3 respectively.

We make the following empirical observations. For f_1 the error appears to decay algebraically, like N^{-4}. For f_2 the error is proportional to e^{-N}, and for f_3 the error is proportional to $e^{-N\log N}$. Typically one observes algebraic, exponential, and super-exponential convergence for C^k, analytic, and entire functions respectively. These convergence rates are similar to those for spectral differentiation. For function f_1 the error in spectral differentiation appears to decay like N^{-2}.

2.2 Spectral integration of analytic functions

The following analysis which leads us to a convergence theorem is based on a proof from a manuscript in preparation by Reddy and Weideman [9].

Let $E_N(f)$ denote the spectral integration error

$$E_N(f) = \left\| \int_{-1}^{x} [f(s) - p_N(s)] ds \right\| \tag{2.4}$$

where $\|\bullet\|$ denotes the maximum norm on [-1,1]. Then

$$\begin{aligned} E_N(f) &\le \left\| \int_{-1}^{x} |f(s) - p_N(s)| ds \right\| \\ &\le \|f - p_N\| \left\| \int_{-1}^{x} ds \right\| \\ &\le 2\|f - p_N\|. \end{aligned}$$

Suppose $f(z)$ is analytic in a simply connected domain D which contains the interval [-1,1]. Let $\Gamma \subset D$ be a regular closed curve enclosing [-1,1]. The polynomial interpolation error is given by Hermite's remainder formula [4]:

$$R_N(x) = f(x) - p_N(x) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\omega_{N+1}(x) f(z)}{\omega_{N+1}(z)(z - x)} dz,$$

where the polynomial $\omega_{N+1}(x)$ is defined as

$$\omega_{N+1}(x) = c \prod_{j=0}^{N} (x - x_j).$$

We note that the normalization constant c cancels in the remainder formula. It can be shown that $\omega_{N+1}(x)$ and $T_{N+1}(x) - T_{N-1}(x)$ are both polynomials of degree $N+1$ with the same roots (see eq. 2.1), thus the two differ by at most a scale factor. For estimating the remainder integral it is convenient to take

$$\omega_{N+1}(x) = T_{N+1}(x) - T_{N-1}(x).$$

Then

$$E_N(f) \le 2\|R_N(x)\| = \frac{1}{\pi} \|\omega_{N+1}(x)\| \left| \int_{\Gamma} \frac{f(z)}{\omega_{N+1}(z)(z - x)} dz \right|.$$

Since $\|T_j(x)\| = 1$ it follows that $\|\omega_{N+1}(x)\| \le 2$, thus we find

(2.5) $$E_N(f) \le \frac{2}{\pi}\left|\int_\Gamma \frac{f(z)}{\omega_{N+1}(z)(z-x)}dz\right|.$$

To obtain a useful upper bound on $E_N(f)$ and the convergence rate we estimate the integral in (2.5). Take the contour Γ to be the ellipse E_ρ centered at 0 with foci ± 1, semi-major axis $\frac{1}{2}(\rho+\rho^{-1})$, and semi-minor axis $\frac{1}{2}(\rho-\rho^{-1})$.

(2.6) $$E_\rho = \left\{z \mid z = \tfrac{1}{2}(\rho e^{i\theta} + \rho^{-1}e^{-i\theta}),\ 0 \le \theta \le 2\pi,\ \rho > 1\right\}$$

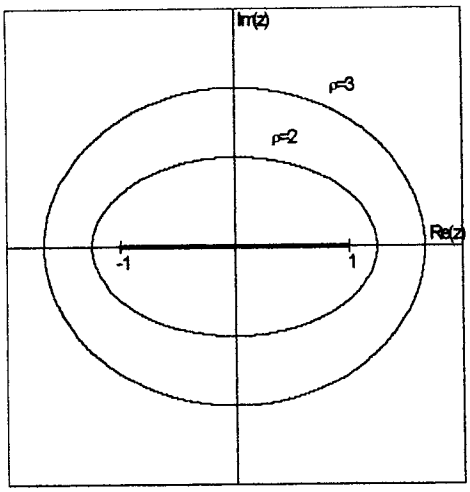

Figure 2.2: E_ρ

This particular contour is used because $\omega_{N+1}(z)$ is of nearly constant modulus on E_ρ for large N. Let $\ell(E_\rho)$ denote the arc length of E_ρ, and C_ρ denote $\max_{z\in E_\rho}|f(z)|$. Then

$$\left|\int_{E_\rho} \frac{f(z)}{\omega_{N+1}(z)(z-x)}dz\right| \le \frac{C_\rho \ell(E_\rho)}{\min_{z\in E_\rho}|\omega_{N+1}(z)(z-x)|}.$$

Rather than expressing $\ell(E_\rho)$ as an elliptic integral of the second kind, we find it convenient to overestimate ℓ by the *rms* approximation $\ell(E_\rho) \le \pi\sqrt{\rho^2+\rho^{-2}}$. By

expressing this in terms of the eccentricity e it can be shown that the maximum relative error occurs as E_ρ degenerates to the interval [-1,1]. The $\lim_{e\to 1} \frac{\ell_{rms}}{\ell} = \frac{\pi}{\sqrt{8}} < 1.12$, thus the *rms* approximation over-estimates $\ell(E_\rho)$ by less than twelve percent.

The modulus of $z - x$ for $z \in E_\rho$ is greater than or equal to the minimum distance from E_ρ to the interval [-1,1]. The nearest approach occurs on the real axis, with the minimum distance $\frac{1}{2}(\rho + \rho^{-1}) - 1$. Thus $|z - x| \geq \frac{1}{2}(\rho + \rho^{-1}) - 1$ for $z \in E_\rho$.

Introducing the variable $w = \rho e^{i\theta}$, $z = \frac{1}{2}(w + w^{-1})$, and $T_N(z) = \frac{1}{2}(w^N + w^{-N})$. Straightforward calculation shows

$$T_{N+1}(z) - T_{N-1}(z) = (w - w^{-1})T_N(z) \qquad z \in E_\rho.$$

To estimate $|\omega_{N+1}(z)|$ note that

$$|T_N(z)| = \tfrac{1}{2}|w^N + w^{-N}| \geq \tfrac{1}{2}(|w^N| - |w^{-N}|) = \tfrac{1}{2}(\rho^N - \rho^{-N}),$$

$$|w - w^{-1}| \geq |w| - |w^{-1}| = \rho - \rho^{-1}.$$

Introducing $\eta > 0$, with $\rho = e^\eta$ we have

$$|T_N(z)| \geq \sinh(\eta N) \quad \text{and} \quad |w - w^{-1}| \geq 2\sinh(\eta) \qquad \text{for } z \in E_\rho.$$

Using the definition of η it follows that

$$|\omega_{N+1}(z)| \geq 2\sinh(\eta)\sinh(\eta N),$$

$$|z - x| \geq \frac{\rho - 2 + \rho^{-1}}{2} = 2\left(\frac{\rho^{\frac{1}{2}} - \rho^{-\frac{1}{2}}}{2}\right)^2 = 2\sinh^2(\eta/2),$$

$$\ell(E_\rho) \leq \pi\sqrt{2\cosh(2\eta)} \qquad \text{for } z \in E_\rho.$$

We can now bound $E_N(f)$, however to show the exponential dependence explicitly we make use of the inequality

$$2\sinh(\eta N) > \frac{2\sinh(\eta N)}{1 + e^{-2\eta N}} = \tanh(\eta N)\, e^{\eta N},$$

which implies

$$\min_{z\in E_\rho} |\omega_{N+1}(z)| > \sinh(\eta)\tanh(\eta N)\, e^{\eta N}.$$

Thus,

$$E_N(f) \; < C_\rho \frac{\sqrt{2\cosh(2\eta)}\,\coth(\eta N)e^{-\eta N}}{\sinh(\eta)\sinh^2(\eta/2)}.$$

Letting $\psi(\eta)$ denote $\dfrac{\sqrt{2\cosh(2\eta)}}{\sinh(\eta)\sinh^2(\eta/2)}$ we have proven the following theorem [9].

THEOREM 1: Suppose $f(z)$ is analytic in some ellipse E_ρ defined by (2.6), with $1 < \rho = e^\eta$. Then, $\forall\, N \geq 1$ the spectral integration error defined by (2.4) satisfies

$$E_N(f) \; < C_\rho \psi(\eta)\coth(\eta N)e^{-\eta N}. \tag{2.7}$$

Remark: The maximum of $C_\rho\psi(\eta)$ depends only on the function f, particularly on the location of any singularities. As the hyperbolic cotangent approaches 1 asymptotically for large values of the argument, we have in (2.7) the reason behind the observed exponential convergence rate for spectral integration of analytic functions. For functions analytic in a domain D the minimum error bound for given N is obtained by minimizing (2.7) with respect to η, subject to the constraint $E_\rho \subset D$. For (non-polynomial) entire functions the minimum error bound for given N is again obtained by minimizing (2.7) only without the constraint on η.

The point of departure in the error analysis for $C^k[-1,1]$ functions is the general formula $E_N(f) \leq 2\|f - p_N\|$, and the interpolation error estimate (equation 9.5.23) [3]

$$\|f - p_N\| \leq CN^{\frac{1}{2}-k}\|f\|_{H^k_w(-1,1)}.$$

Here $\|f\|_{H^k_w(-1,1)}$ denotes the weighted Sobolev norm

$$\|f\|_{H^k_w(-1,1)} = \left(\sum_{m=0}^{k}\int_{-1}^{1}|f^{(m)}(x)|^2 w(x)dx\right)^{\frac{1}{2}},$$

with weight function $w(x) = (1-x^2)^{-\frac{1}{2}}$.

Since $C^k[-1,1] \subset H^k(-1,1)$, for $f \in C^k[-1,1]$ we obtain the estimate:

$$\|f - p_N\| \leq CN^{\frac{1}{2}-k} \|f\|_{H^k_w(-1,1)},$$
$$E_N(f) \leq CN^{\frac{1}{2}-k} \|f\|_{H^k_w(-1,1)}.$$

If in addition $f \in H^{k+1}_w(-1,1)$ then

$$E_N(f) \leq CN^{-k-\frac{1}{2}} \|f\|_{H^{k+1}_w(-1,1)}.$$

We note that these results are not sharp. For example, the function $f_1(x) = |x|^3$ considered in Section 2.1 is an element of $H^3_w(-1,1)$ and the preceding analysis would suggest the spectral integration error decays like $N^{-2.5}$ while the actual error is found to decay like N^{-4}.

2.3 Efficient computation of the spectral integration matrices

As pointed out in section 2.1, generating the integration matrices directly from the definition (2.3) is not particularly efficient. The brute force approach requires $O(N^3)$ operations. This section is devoted to developing a more efficient procedure. The algorithm we present reduces this to $O(N^2 \log N)$ operations by exploiting the FFT. The basis for the procedure lies in a relationship between the discrete Fourier and Chebyshev transforms. Certain matrix symmetry properties are used to further reduce the operation count. A result (which we believe may be new) on the spectrum of the inverse Chebyshev transform matrix is presented. This result implies stability of the forward and inverse Chebyshev transforms.

The discussion is greatly simplified by introducing the following notation. Let $\boldsymbol{T}$ denote the inverse Chebyshev transform (inverse DCT) matrix $T_{ij} = T_j(x_i) = \cos(ij\pi/N)$, $i,j = 0,1,...,N$. Note that $\boldsymbol{T}$ is symmetric and observe the following additional symmetries. The $0th, 2nd, ..., Nth$ columns are symmetric, while the $1st, 3rd, ..., (N-1)st$

columns are anti-symmetric. That is $T_{N-i,j} = (-1)^j T_{ij}$, a consequence of the fact that $T_j(x)$ is an even (resp. odd) function of x for j even (resp. odd).

Let $\boldsymbol{U}$ denote the product $\boldsymbol{T}\hat{\boldsymbol{A}}$. As $\hat{\boldsymbol{A}}$ has only $2N$ nonzero entries, $\boldsymbol{U}$ can be formed explicitly in $O(N^2)$ operations. The columns of $\boldsymbol{U}$ also have alternating symmetry which we express as $U_{N-i,j} = (-1)^{j+1} U_{ij}$. Thus $\boldsymbol{U}$ has at most $\frac{N^2+2N}{2}$ distinct entries, assuming that N is even. The columns of $\boldsymbol{U}$ may be viewed as antiderivatives of the Chebyshev polynomials sampled at the Chebyshev points.

With $\boldsymbol{T}$ and $\boldsymbol{U}$ defined as in the previous paragraphs we have the linear matrix equation $\boldsymbol{AT} = \boldsymbol{U}$ for the antidifferentiation matrix $\boldsymbol{A}$.

Lemma 1: $\boldsymbol{A}$ is anti-centrosymmetric (*ACS*), $A_{ij} = -A_{N-i,N-j} \quad i,j = 0,1,\ldots,N.$

The proof proceeds by constructing the *ACS* part of $\boldsymbol{A}$ and taking the product with $\boldsymbol{T}$ to obtain $\boldsymbol{U}$.

Proof:

$$\begin{aligned}
\sum_{k=0}^{N} \frac{A_{ik} - A_{N-i,N-k}}{2} T_{kj} &= \tfrac{1}{2} \sum_{k=0}^{N} A_{ik} T_{kj} - \tfrac{1}{2} \sum_{k=0}^{N} A_{N-i,N-k} T_{kj} \\
&= \tfrac{1}{2} U_{ij} - \tfrac{1}{2} \sum_{m=0}^{N} A_{N-i,m} T_{N-m,j} \\
&= \tfrac{1}{2} U_{ij} - \tfrac{1}{2} \sum_{m=0}^{N} A_{N-i,m} (-1)^j T_{mj} \\
&= \tfrac{1}{2} U_{ij} - \tfrac{(-1)^j}{2} U_{N-i,j} \\
&= \tfrac{1}{2} U_{ij} - \tfrac{(-1)^j}{2} (-1)^{j+1} U_{ij} \\
&= U_{ij}
\end{aligned}$$

This implies $A_{ik} = -A_{N-i,N-k}$, thus $\boldsymbol{A}$ is *ACS* as claimed. □

Remark: This property is of interest because of the implications for numerical computation. We have shown that the *ACS* property of $\boldsymbol{A}$ is a consequence of the symmetry properties of $\boldsymbol{U}$. Thus the information contained in the lower $N/2$ rows of $\boldsymbol{U}$ is superfluous. With this in mind we let $\widetilde{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{U}}$ denote the upper $(N/2+1)\times(N+1)$ submatrices of $\boldsymbol{A}$ and $\boldsymbol{U}$ respectively, implicitly assuming N is even. We have thus reduced the problem to

$$\widetilde{\boldsymbol{A}}\boldsymbol{T}=\widetilde{\boldsymbol{U}} \quad \Rightarrow \quad \widetilde{\boldsymbol{A}}^{\mathrm{T}}=\boldsymbol{T}^{-1}\widetilde{\boldsymbol{U}}^{\mathrm{T}} \text{ or } \widetilde{\boldsymbol{A}}^{\mathrm{T}}=\boldsymbol{C}\widetilde{\boldsymbol{U}}^{\mathrm{T}} \tag{2.8}$$

where $\boldsymbol{C}$ is the DCT matrix.

Before proceeding we record two properties of the centro-transpose. Suppose $\boldsymbol{B}$ and $\boldsymbol{C}$ are arbitrary square matrices of like dimension, then

(i) $(\boldsymbol{B}^{C})^{C}=\boldsymbol{B}$
(ii) $(\boldsymbol{BC})^{C}=\boldsymbol{B}^{C}\boldsymbol{C}^{C}$.

An immediate consequence of the anti-centrosymmetry of $\boldsymbol{A}$ is

$$\boldsymbol{S}_r=-\boldsymbol{P}^{C}\boldsymbol{A}=(\boldsymbol{PA})^{C}=\boldsymbol{S}_I^{C}. \tag{2.9}$$

Thus $\boldsymbol{S}_r$ is obtained from $\boldsymbol{S}_I$ by a simple re-indexing, which involves only fixed point arithmetic.

Next we consider the matter of expedient computation of the DCT. The basic idea is to take advantage of the relationship between the discrete Fourier transform (DFT) and discrete Chebyshev transforms, thereby exploiting the efficiency of the FFT. The $2N$ point DFT is defined by

$$\widetilde{f}_k=\sum_{j=0}^{2N-1} f_j \exp(-2\pi ijk/2N) \qquad k=0,1,\ldots,2N-1.$$

The DCT is defined by

$$\widehat{f_k} = \frac{2}{Nc_k}\sum_{j=0}^{N}\frac{f_j}{c_j}\cos(\pi jk/N),\ k = 0,1,...,N,\ c_i = \begin{cases} 2 & i = 0,N \\ 1 & 1 \le i \le N-1 \end{cases}.$$

To derive the relathionship between the DCT and DFT let us define the sequence

$$\{g_j\}_{j=0}^{2N-1} \text{ where } g_j = \begin{cases} f_j & j = 0,1,...,N \\ f_{2N-j} & j = N+1,N+2,...,2N-1. \end{cases}$$

The DFT of $\{g\}$ is

$$\begin{aligned}\widetilde{g_k} &= \sum_{j=0}^{2N-1} g_j \exp(-2\pi ijk/2N) \\ &= g_0 + \sum_{j=1}^{N-1} g_j \exp(-\pi ijk/N) + g_N \cos(Nk\pi/N) + \sum_{j=N+1}^{2N-1} g_j \exp(-\pi ijk/N) \\ &= f_0 + \sum_{j=1}^{N-1} f_j \exp(-\pi ijk/N) + f_N \cos(Nk\pi/N) + \sum_{j=N+1}^{2N-1} g_j \exp(-\pi ijk/N).\end{aligned}$$

Making the change of index $m = 2N - j$ in the second sum

$$\sum_{j=N+1}^{2N-1} g_j \exp(-\pi ijk/N) = \sum_{m=1}^{N-1} g_{2N-m} \exp(-\pi i(2N-m)k/N) = \sum_{m=1}^{N-1} f_m \exp(\pi imk/N).$$

Then

$$\begin{aligned}\widetilde{g_k} &= f_0 + \sum_{j=1}^{N-1} f_j \exp(-\pi ijk/N) + f_N \cos(Nk\pi/N) + \sum_{m=1}^{N-1} f_m \exp(\pi imk/N) \\ &= f_0 + 2\sum_{j=1}^{N-1} f_j \cos(\pi jk/N) + f_N \cos(Nk\pi/N) \\ &= 2\sum_{j=0}^{N}\frac{f_j}{c_j}\cos(\pi jk/N).\end{aligned}$$

Thus we have established the relationship $\widehat{f_k} = \frac{1}{Nc_k}\widetilde{g_k}$ for $k = 0,1,...,N.$

Given the sequence f_j $j = 0,1,...,N,$ the DCT may be evaluated in $O(N\log N)$ flops by the following procedure.

- Form the sequence $f_0, f_1, ..., f_{N-1}, f_N, f_{N-1}, f_{N-2}, ..., f_1$.
- Compute the DFT using the FFT algorithm.
- Extract the first $N+1$ components of the transform.

- Divide by N and halve the first and last components.

Given the sequence $\widehat{f_j}$ $j = 0,1,...,N$, the inverse DCT can be evaluated by a similar procedure.

- Form the sequence $2\widehat{f_0}, \widehat{f_1}, ..., \widehat{f_{N-1}}, 2\widehat{f_N}, \widehat{f_{N-1}}, ..., \widehat{f_1}$.
- Compute the DFT using the FFT algorithm.
- Extract the first $N+1$ components of the result.
- Divide the result by 2.

Consider the projector $\boldsymbol{P}$. As $\boldsymbol{P}$ is a rank one perturbation of the identity, it can be applied to $\boldsymbol{A}$ in $2(N+1)^2$ operations as follows. Let $(1,...,1)^T$ be an $N+1$ vector, and let A_N denote the *Nth* row of $\boldsymbol{A}$. Then $\boldsymbol{PA} = \boldsymbol{A} - (1,...,1)^T A_N$.

Uniting these concepts gives the following procedure.

Algorithm for generating $\boldsymbol{S_l}$ and $\boldsymbol{S_r}$ (N even):

- Form $\widetilde{\boldsymbol{U}}$: the upper $N/2 + 1$ rows of $\boldsymbol{U}$.
- Compute $\widetilde{\boldsymbol{A}}^T$: the DCT of $\widetilde{\boldsymbol{U}}^T$.
- Construct $\boldsymbol{A}$ from $\widetilde{\boldsymbol{A}}$ via *ACS* property.
- Obtain $\boldsymbol{S_l}$ from $\boldsymbol{A}$ by projection.
- $\boldsymbol{S_r} = \boldsymbol{S_l^C}$

This algorithm is implemented in the following MATLAB function.

```
function [Sl,Sr] = INTMAT(N)

% Spectral integration matrices of order N+1 (N even)
j=[0:N/2]';

% Form Ũ: the upper N/2+1 rows of U
Ũ(:,1:2)=[cos(j*pi/N),cos(j*pi/N).^2/2];
Ũ(:,3:N+1)=(cos(j*[3:N+1]*pi/N)./(ones(N/2+1,1)*[3:N+1])...
-cos(j*[1:N-1]*pi/N)./(ones(N/2+1,1)*[1:N-1]))/2;

% DCT of Ũ'
B=real(fft([Ũ,Ũ(:,N:-1:2)]'))';
Ã=[B(1:N/2+1,1)/2,B(1:N/2+1,2:N),B(1:N/2+1,N+1)/2]/N;

% Construct A from ACS property
A=[Ã;-flipud(fliplr(Ã(1:N/2,:)))];

% Projection and centro-transpose
Sl=A-ones(N+1,1)*A(N+1,:);
Sr=fliplr(flipud(Sl));
```

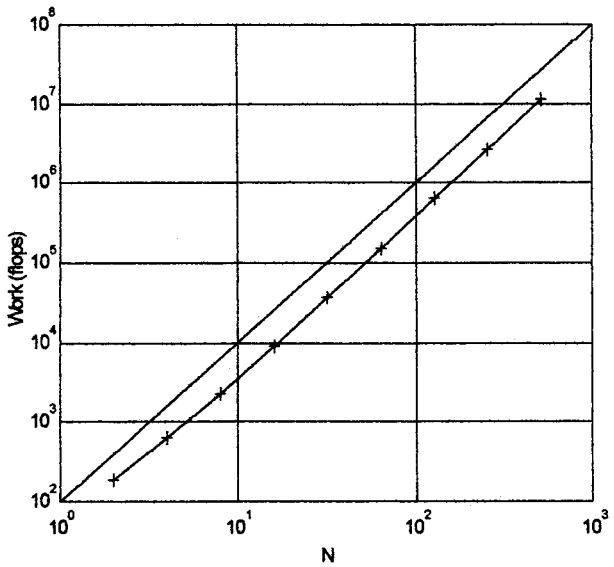

Figure 2.3 Work (flops): function INTMAT. Compare the work characteristic with the solid line ($100N^2$). We note that this is an $N^2 \log N$ algorithm, but for practical purposes the work scales like N^2 for N in the range 2 through 512.

Next we consider the condition number of $\boldsymbol{T}$ with respect to inversion.

THEOREM 2: Let $\boldsymbol{T}$ denote the matrix with entries $T_{ij} = \cos(ij\pi/\mathrm{N}) \quad i,j = 0,1,2,...,N$ with $N \geq 4$, N even. Then

$$\kappa_2(\boldsymbol{T}) = \left[2 + \frac{1+\sqrt{4N+1}}{N} \right]^{\frac{1}{2}}. \tag{2.10}$$

As the proof is rather lengthy, we provide only a sketch of the details. Consider $\boldsymbol{T}^2$, it can be shown that $(T^2)_{00} = (T^2)_{\mathrm{NN}} = N+1$, and all other main diagonal entries equal $N/2+1$. One can also show that the $1st, 3rd, ..., (N-1)st$ (sub and super) diagonals consist of zeros at each position, and the $2nd, 4th, ...Nth$ (sub and super) diagonals consist of ones at each entry. Proceed by forming the characteristic equation $\det(\boldsymbol{T}^2 - \lambda \boldsymbol{I}) = 0$. Perform elementary row operations to obtain an upper triangular determinant with the factors of the characteristic polynomial on the diagonal. One then obtains the spectrum

of $\boldsymbol{T}^2$:

$\lambda_1 = N/2$ of algebraic multiplicity $N-3$

$\lambda_2 = N$ algebraic multiplicity 2

$\lambda_3 = N + \frac{1}{2} - \sqrt{N + \frac{1}{4}}$

$\lambda_4 = N + \frac{1}{2} + \sqrt{N + \frac{1}{4}}$

$\boldsymbol{T}^2$ is *SPD* which implies $\kappa_2(\boldsymbol{T}^2) = \lambda_4/\lambda_1$. Finally, symmetry of $\boldsymbol{T}$ implies $\kappa_2(\boldsymbol{T}) = \sqrt{\kappa_2(\boldsymbol{T}^2)}$ which completes the proof. We emphasize that $\kappa_2(\boldsymbol{T}) \to \sqrt{2}$ as $N \to \infty$, thus the DCT matrix is well-conditioned (recall $\kappa = 1$ is ideal). The obvious implication is that the DCT and inverse transform are stable with respect to perturbations in the data.

We have found that Kauthen's algorithm is limited to low order polynomial expansions by the use of the standard basis $\{1,x,...,x^N\}$ for $\boldsymbol{P}_N$. In his algorithm the matrix, which corresponds to $\boldsymbol{S}_I$ in this paper, is determined by requiring the integration be exact for the monomials $1,x,x^2,...,x^N$. This leads to a linear system involving a Vandermonde matrix. Tyrtyshnikov [12] has shown that Vandermonde systems with distinct nodes $|x_j| \leq 1$ are exponentially ill-conditioned. The condition number grows exponentially with the order of the matrix. For N as small as 45 the spectral condition number of this Vandermonde matrix exceeds 10^{16}. Thus accurate solutions are not obtained for large N.

2.4 Matrix-free implementation with the FFT

As the spectral integration approach results in dense collocation matrices, Gaussian elimination is not the most attractive option for obtaining solutions to linear problems when N is large. The alternative is an iterative approach. To apply iterative methods

efficiently, it is highly desirable to be able to apply the spectral integration operators without explicitly generating and performing computations with large matrices. In this section we show how this can be accomplished with the Fast Fourier Transform.

Matrix-free algorithm for the computation of $\boldsymbol{S}_I \boldsymbol{f} = \boldsymbol{P}\boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}\boldsymbol{C}\boldsymbol{f}$:

- Compute the DCT of f via the FFT: $O(N\log N)$ operations.
- Apply $\widehat{\boldsymbol{A}}$: $O(N)$ operations.
- Inverse transform via the FFT: $O(N\log N)$ operations.
- Projection: $O(N)$ operations.

Note the overall operation count scales like $N\log N$. This algorithm is implemented in the following MATLAB function.

```
function FL=S_L(f,N)

% FFT based Spectral Integration
% allocate memory
b=zeros(N+1,1);

% DCT
a=fft([f;f(N:-1:2)])/N;
a(1)=a(1)/2;a(N+1)=a(N+1)/2;

% Antidifferentiate
b(1)=a(2)/4;
b(2)=a(1)-a(3)/2;
b(3:N-1)=(a(2:N-2)-a(4:N))./[4:2:2*N-4]';
b(N)=a(N-1)/(2*N-2)-a(N+1)/(N*N-1);
b(N+1)=a(N)/(2*N);

% Inverse Transform
c=[b;b(N:-1:2)];
c(1)=c(1)*2;c(N+1)=c(N+1)*2;
%remove imaginary part introduced by rounding errors
F=real(fft(c))/2;

% Projection
FL=F(1:N+1)-ones(N+1,1)*F(N+1);
```

We compare the efficiency of the matrix-free and standard matrix-vector product implementations. Figure 2.4 shows the computational work (flops counted by MATLAB) required to apply $\boldsymbol{S}_I$. Results are shown for $N = 8, 16, 32, \ldots, 512$.

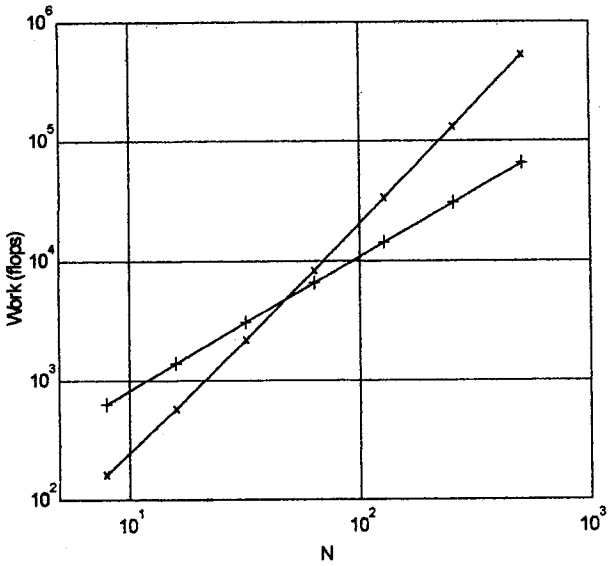

Figure 2.4. Work (flops) to apply S_I : matrix vs. transform based integration. The curve for FFT based integration is marked by '+' signs, the operation count is approximately $42N + 14N\log N$. In the matrix case, only the work to apply the matrix is counted.

The cross-over point is slightly less than $N = 64$, for larger N the transform based implementation offers considerable savings. An additional benefit of the matrix-free approach is the substantial reduction in memory usage. As we envision applying spectral integration to problems in more than one dimension, the efficiency offered by fast transform methods may prove to be very significant.

Chapter 3
Integral Formulation and Discretization

3.1 Integral formulation of boundary value problems

We consider linear two-point boundary value problems posed on the interval [-1,1] with separated Dirichlet boundary conditions

$$u'' + p(x)u' + q(x)u = r(x), \qquad u(-1) = \alpha, \qquad u(1) = \beta \tag{3.1}$$

assuming $p, q,$ and r are bounded integrable functions with p at least once differentiable. We apply a standard transformation [13] to equation (3.1) to obtain an integral formulation of the problem. The transformation consists of multiplying the differential equation by the Green's function for the Poisson equation $u'' = f(x),\ u(\pm 1) = 0,$ and integrating with respect to s over [-1,1]. As mentioned in Chapter 1 this Green's function is

$$G(x,s) = \begin{cases} (x-1)(s+1)/2 & -1 \le s \le x \\ (x+1)(s-1)/2 & x \le s \le 1 \end{cases}.$$

We define the linear integral operator K by

$$Kf(x) = \int_{-1}^{1} G(x,s)f(s)ds,$$

which for our purposes is conveniently expressed as

$$Kf(x) = \frac{x-1}{2}\int_{-1}^{x}(s+1)f(s)ds + \frac{x+1}{2}\int_{x}^{1}(s-1)f(s)ds. \tag{3.2}$$

With this notation the transformation of (3.1) is written

$$K(u'' + pu' + qu) = Kr. \tag{3.3}$$

Integrating the Ku'' term by parts

$$Ku''(x) = \frac{x-1}{2}\int_{-1}^{x}(s+1)u''(s)ds + \frac{x+1}{2}\int_{x}^{1}(s-1)u''(s)ds$$

$$= \frac{x-1}{2}\left[(s+1)u'(s)\Big|_{-1}^{x} - \int_{-1}^{x}u'(s)ds\right] + \frac{x+1}{2}\left[(s-1)u'(s)\Big|_{x}^{1} - \int_{x}^{1}u'(s)ds\right]$$

$$= \frac{x-1}{2}[(x+1)u'(x) - (u(x)-\alpha)] + \frac{x+1}{2}[-(x-1)u'(x) - (\beta - u(x))],$$

we obtain

$$Ku'' = u(x) + \frac{\alpha-\beta}{2}x - \frac{\alpha+\beta}{2}. \tag{3.4}$$

The $K(pu')$ term is integrated by parts to shift the derivative to p (the reason for the differentiability assumption on p):

$$K(pu') = \frac{x-1}{2}\int_{-1}^{x}(s+1)p(s)u'(s)ds + \frac{x+1}{2}\int_{x}^{1}(s-1)p(s)u'(s)ds$$

$$= \frac{x-1}{2}\left[(s+1)p(s)u(s)\Big|_{-1}^{x} - \int_{-1}^{x}[p(s)+(s+1)p'(s)]u(s)ds\right]$$

$$+ \frac{x+1}{2}\left[(s-1)p(s)u(s)\Big|_{x}^{1} - \int_{x}^{1}[p(s)+(s-1)p'(s)]u(s)ds\right]$$

$$= \frac{x-1}{2}\left[(x+1)p(x)u(x) - \int_{-1}^{x}p(s)u(s)ds - \int_{-1}^{x}(s+1)p'(s)u(s)ds\right]$$

$$+ \frac{x+1}{2}\left[-(x-1)p(x)u(x) - \int_{x}^{1}p(s)u(s)ds - \int_{x}^{1}(s-1)p'(s)u(s)ds\right].$$

Defining the integral operator J by

$$Jf(x) = \frac{x-1}{2}\int_{-1}^{x}f(s)ds + \frac{x+1}{2}\int_{x}^{1}f(s)ds, \tag{3.5}$$

we have

(3.6) $$K(pu') = -J(pu) - K(p'u).$$

Taking the boundary terms $\frac{\alpha-\beta}{2}x - \frac{\alpha+\beta}{2}$ from (3.4) to the right hand side of (3.3), let

$$f(x) = Kr(x) + \frac{\alpha+\beta}{2} + \frac{\beta-\alpha}{2}x.$$

The linear Fredholm integral equation equivalent to the boundary value problem (3.1) is then

(3.7) $$u(x) = J(pu) + K((p' - q)u) + f(x).$$

This transformation is also applicable to nonlinear differential equations as long as the integrations necessary to remove the derivatives on u can be performed, as in the case of the quasilinear equation $u'' + p(x)u' = f(x,u)$.

3.2 Discrete analogues of the operators K and J

The idea behind our discretization scheme is to approximate the integrals in the K and J operators by spectral integration on the Chebyshev points.

Consider $Kf(x)$, evaluating at $x = x_j$ we have

(3.8) $$Kf(x_j) = \frac{x_j - 1}{2}\int_{-1}^{x_j}(s+1)f(s)ds + \frac{x_j+1}{2}\int_{x_j}^{1}(s-1)f(s)ds.$$

In order to approximate these two integrals by the spectral integration operators S_l and S_r we sample the two functions $(s+1)f(s)$ and $(s-1)f(s)$ at the Chebyshev points s_j, $j = 0,1,...N$. Since the two sets $\{s_j\}_{j=0}^{N}$ and $\{x_j\}_{j=0}^{N}$ are identical we take the liberty of referring to either as simply $\{x_j\}$. As in Chapter 2 we let $\boldsymbol{x} = (x_0,x_1,...,x_N)^T$ and $\boldsymbol{f} = (f(x_0),f(x_1),...,f(x_N))^T$. Then

$$(\boldsymbol{X} + \boldsymbol{I})\boldsymbol{f} = ((x_0+1)f(x_0),...,(x_N+1)f(x_N))^T$$

and

$$(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{f} = ((x_0-1)f(x_0),\ldots,(x_N-1)f(x_N))^T,$$

where $\boldsymbol{X} = diag(\boldsymbol{x})$ and $\boldsymbol{I}$ is the $(N+1)\times(N+1)$ identity matrix. We now express the approximation of the integrals in (3.8) as

$$[\boldsymbol{S}_l(\boldsymbol{X}+\boldsymbol{I})\boldsymbol{f}]_j \approx \int_{-1}^{x_j}(s+1)f(s)ds, \text{ and } [\boldsymbol{S}_r(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{f}]_j \approx \int_{x_j}^{1}(s-1)f(s)ds.$$

Thus we define $\boldsymbol{K}_N$, the discrete analogue of the continuous operator K, by

$$\boldsymbol{K}_N = \frac{(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{S}_l(\boldsymbol{X}+\boldsymbol{I}) + (\boldsymbol{X}+\boldsymbol{I})\boldsymbol{S}_r(\boldsymbol{X}-\boldsymbol{I})}{2}. \tag{3.9}$$

Where $\boldsymbol{K}_N\boldsymbol{f}$ approximates $(Kf(x_0),Kf(x_1),\ldots Kf(x_N))^T$, the approximation being exact in the case that f is a polynomial of degree $N-1$ or less. Note the analogous forms of (3.9) and (3.2).

We note that $\boldsymbol{X}$ is ACS. This follows from $X_{jj} = x_j$, and the fact that

$$x_{N-j} = \cos((N-j)\pi/N) = \cos(\pi - j\pi/N) = -\cos(j\pi/N) = -x_j.$$

Thus $$(\boldsymbol{X}-\boldsymbol{I})^C = -(\boldsymbol{X}+\boldsymbol{I}) \quad \text{and} \quad (\boldsymbol{X}+\boldsymbol{I})^C = -(\boldsymbol{X}-\boldsymbol{I}).$$

It then follows from (2.9) and (3.9) that $\boldsymbol{K}_N$ is the centro-symmetric (CS) part of $(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{S}_l(\boldsymbol{X}+\boldsymbol{I})$, therefore $\boldsymbol{K}_N$ is CS. This CS property is primarily of interest to us because it may be exploited to simplify computations.

In the same spirit we define $\boldsymbol{J}_N$, the discrete analogue of the continuous operator J,

$$\boldsymbol{J}_N = \frac{(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{S}_l + (\boldsymbol{X}+\boldsymbol{I})\boldsymbol{S}_r}{2}. \tag{3.10}$$

Noting that $\boldsymbol{J}_N$ is ACS, in fact, the ACS part of $(\boldsymbol{X}-\boldsymbol{I})\boldsymbol{S}_l$.

To illustrate the use of the discrete operators we consider a simple model problem

$$u'' + au' + bu = r(x), \quad u(-1) = \alpha, \quad u(1) = \beta, \quad a, b \text{ constants.}$$

Recalling (3.4) $Ku''(x) = u(x) + \frac{\alpha - \beta}{2}x - \frac{\alpha + \beta}{2}$ and (3.6) $Ku'(x) = -Ju(x)$ we reformulate the boundary value problem as an integral equation

$$u - aJu + bKu = Kr + \frac{\alpha + \beta}{2} + \frac{\beta - \alpha}{2}x.$$

Or upon letting $f(x)$ denote $Kr(x) + \frac{\alpha + \beta}{2} + \frac{\beta - \alpha}{2}x$,

$$u(x) - aJu(x) + bKu(x) = f(x).$$

We then have the analogous discrete equation

$$[\boldsymbol{I} - a\boldsymbol{J} + b\boldsymbol{K}]\boldsymbol{u} = \boldsymbol{f}.$$

This linear algebraic system is then solved for $\boldsymbol{u}$, the approximate solution of the boundary value problem. In the case of variable coefficient equations, $u'' + p(x)u' + q(x)u = r(x)$, the equivalent linear system is

$$[\boldsymbol{I} - \boldsymbol{JP} + \boldsymbol{K}(\boldsymbol{Q} - \boldsymbol{P}')]\boldsymbol{u} = \boldsymbol{f}.$$

Where $\boldsymbol{P} = diag(p(x_0), p(x_1), \ldots p(x_N))$ with $\boldsymbol{P}'$ and $\boldsymbol{Q}$ defined analogously.

3.3 The treatment of Neumann and Robin boundary conditions

The formalism developed for Dirichlet problems in the preceding sections may be generalized to include both Neumann and Robin type boundary conditions. Consider the linear boundary value problem with Dirichlet condition at $x = -1$ and Robin condition at $x = 1$,

$$L[u] = f(x), \quad u(-1) = \alpha, \quad au(1) + u'(1) = \beta.$$

A Neumann condition being obtained in the case $a = 0$. Assuming that $a \neq -1/2$, the Green's function for the problem $u''(x) = f(x), \quad u(-1) = \alpha, \quad au(1) + u'(1) = \beta$, is

$$G(x,s) = \frac{1}{2a+1}\begin{cases} (s+1)(ax-1-a) & -1 \le s \le x \\ (x+1)(as-1-a) & x \le s \le 1 \end{cases}.$$

This procedure breaks down in the case $a = -1/2$. Any function of the form $c(x+1)$, where c is a constant, satisfies $u'' = 0$, $u(-1) = 0$, $-\frac{1}{2}u(1) + u'(1) = 0$. This implies that

$$u''(x) = f(x), \quad u(-1) = \alpha, \quad -\tfrac{1}{2}u(1) + u'(1) = \beta$$

does not admit a unique solution. If u_1 is any solution, then $u_1 + c(x+1)$ is also a solution for any constant c. The consequence of this non-uniqueness is that we cannot define a Green's function, thus we must exclude the case $a = -1/2$ in this approach. An alternative approach is discussed at the end of this section.

An integration by parts yields

$$\int_{-1}^{1} G(x,s)u''(s)ds = u(x) + \frac{a\alpha - \beta}{2a+1}x - \frac{(a+1)\alpha + \beta}{2a+1}. \tag{3.11}$$

Which in the case $a = 0$ reduces to

$$\int_{-1}^{1} G(x,s)u''(s)ds = u(x) - \beta x - \alpha - \beta. \tag{3.12}$$

Similarly

$$\int_{-1}^{1} G(x,s)u'(s)ds = -\frac{ax-1-a}{2a+1}\int_{-1}^{x} u(s)ds - \frac{x+1}{2a+1}\left[u(1) + a\int_{x}^{1} u(s)ds \right], \tag{3.13}$$

or with a Neumann condition

$$\int_{-1}^{1} G(x,s)u'(s)ds = \int_{-1}^{x} u(s)ds - (x+1)u(1). \tag{3.14}$$

Note that $u(1)$ obtained from the integral of u' is an unknown quantity. To illustrate how this undetermined boundary value is treated, we consider the discretization of the

model problem

$$u'' + u' = 0, \quad u(-1) = \alpha, \quad au(1) + u'(1) = \beta. \tag{3.15}$$

Referring to (3.11) and (3.13) and letting $f(x) = \alpha(a+1) + \beta + (\beta - \alpha a)x$, we obtain the equivalent integral equation

$$(2a+1)u(x) + (a+1-ax)\int_{-1}^{x} u(s)ds - a(x+1)\int_{x}^{1} u(s)ds - (x+1)u(1) = f(x).$$

Letting $\boldsymbol{e}_0 = (1,0,...,0)^T$, $\boldsymbol{i} = (1,...,1)^T$, and $\boldsymbol{x} = (x_0,x_1,...,x_N)^T$ we have the corresponding discrete equation

$$[(2a+1)\boldsymbol{I} + ((a+1)\boldsymbol{I} - a\boldsymbol{X})\boldsymbol{S}_l - a(\boldsymbol{X}+\boldsymbol{I})\boldsymbol{S}_r - (\boldsymbol{x}+\boldsymbol{i})\boldsymbol{e}_0^T]\boldsymbol{u} = \boldsymbol{f},$$

the $(\boldsymbol{x}+\boldsymbol{i})\boldsymbol{e}_0^T\boldsymbol{u}$ term corresponding to the $(x+1)u(1)$ term in the integral equation.

Variable coefficient equations are treated as in the Dirichlet / Dirichlet case (assuming $p(x)$ is differentiable), the integration by parts introducing a $p'(s)$ term. An alternate approach to Robin type boundary conditions for the case $a = -\frac{1}{2}$ would be to use a Green's function for an equation which does admit a unique solution. One possibility is to use the Green's function for the problem

$$u'' + u = f(x), \quad u(-1) = 0, \quad au(1) + u'(1) = 0.$$

However the practicality of this approach for numerical computation has not been determined.

Chapter 4
The Solution of Boundary Value Problems: Numerical Examples

4.1 An elementary constant coefficient equation.

For our first example we consider a constant coefficient equation which depends on a real parameter μ

$$u'' - \mu^2 u = 0, \quad u(-1) = 0, \quad u(1) = 1, \tag{4.1}$$

with solution $u(x) = \dfrac{\sinh(\mu(x+1))}{\sinh(2\mu)}$. We solve this problem numerically with $\mu = 1$ and $\mu = 100$, comparing the results obtained by the spectral integration and spectral differentiation approaches.

In the case $\mu = 1$ we have a well-conditioned boundary value problem which does not present any difficulties for a numerical solution. However this problem does serve to illustrate a number of important points. The first point is that for given N the two approaches achieve comparable accuracy. Second, matrix conditioning effects the accuracy of the solution to the boundary value problem. Third, the Gaussian elimination method for solving linear systems is more expensive than an iterative approach, and preconditioning improves the efficiency of iterative methods.

For both approaches the matrix implementation was used and the resulting linear systems solved by Gaussian elimination as implemented by the MATLAB 'backslash' routine. Figure 4.1.1 shows the error norm as a function of N. Here and in the rest of this chapter all vector norms are infinity norms; $\|error\| = \max_{0 \le j \le N} |u(x_j) - u_j|$. Note the degradation in the accuracy of the solution for N greater than 16. For both sets of solutions we have computed the growth factors for LU factorization to all be

approximately 1 [6]. This implies that the observed growth in the error is not due to instability of the Gaussian elimination. In the spectral integration case we believe this is due to rounding errors, since $\kappa_2(\boldsymbol{I}-\boldsymbol{K}_N)$ is essentially constant. We attribute the greater degradation in the differentiation case to the growth of $\kappa_2(\boldsymbol{D}^{(2)}-\boldsymbol{I})$. These condition numbers are given in the following table.

Table 4.1 Condition numbers.

N	$\kappa_2(\boldsymbol{I}-\boldsymbol{K}_N)$	$\kappa_2(\boldsymbol{D}^{(2)}-\boldsymbol{I})$
4	1.4335	6.06
8	1.4088	63.8
16	1.4095	9.40e+2
32	1.4087	1.48e+4
64	1.4078	2.35e+5
128	1.4073	3.76e+6
256	1.4070	6.01e+7

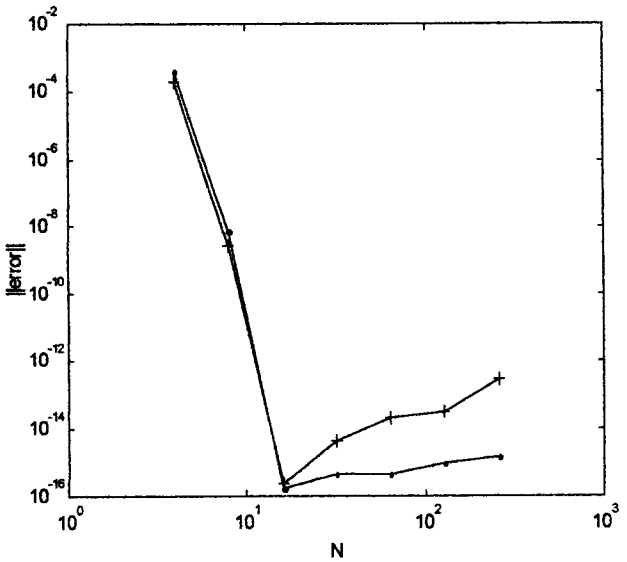

Figure 4.1.1 Error in the solution of (4.1) with $\mu = 1$.
Spectral integration (•), spectral differentiation (+).

Next we compare the efficiency of these two methods in the solution of (4.1). Figure 4.1.2 shows the error as a function of the computational work (flops) to generate the appropriate matrices and perform the Gaussian elimination.

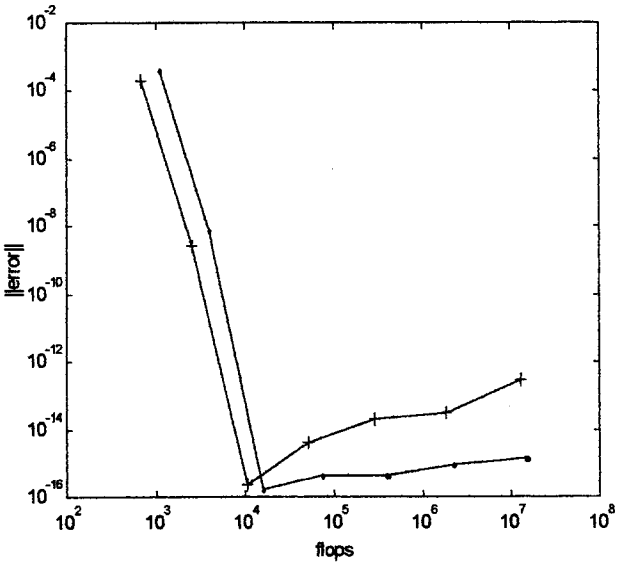

Figure 4.1.2 Efficiency of spectral methods in the solution of (4.1).
The case $\mu = 1$.Linear systems solved by Gaussian elimination.
Spectral integration (•), spectral differentiation (+).

The difference in the number of operations for the two methods is due to two factors. First, $\boldsymbol{D}^{(2)}$ is computed somewhat more efficiently than $\boldsymbol{K}_N$. Second, the linear systems being solved are of dimensions $N+1$ and $N-1$ respectively in the integral and differential approaches.

Next we consider solving these same linear systems by iterative methods. The iterative method employed is the CGS or conjugate gradient squared algorithm [11]. For the spectral differentiation system a preconditioned CGS iteration is employed. We note that without preconditioning the CGS iteration fails to converge to within the

tolerance of 10^{-16} in N iterations. For details on the CGS algorithm and the finite-difference preconditioner refer to the Appendices. Both iterative solutions are implemented without matrices by using fast-transform routines to apply the $\boldsymbol{K}_N$ and $\overline{\boldsymbol{D}}_N^{(2)}$ operators. Figures 4.1.3 and 4.1.4 below show the error as a function of N and error as a function of the computational work. As with the Gaussian elimination solutions the computations were performed with N = 4,8,16,...,256. In comparing Figures 4.1.1 and 4.1.3 we note that the degrading effects of ill-conditioning and roundoff for $N > 16$ are significantly reduced in the iterative solutions. Noting the different horizontal scales in Figures 4.1.2 and 4.1.4 reveals that for $N > 64$ the iterative solutions require considerably less work for comparable N than the direct solutions. The number of CGS iterations for both approaches, as well as condition numbers $\kappa_2(\boldsymbol{M}^{-1}(\overline{\boldsymbol{D}}_N^{(2)} - \boldsymbol{I}))$ of the preconditioned matrices are given in Table 4.2 following Figure 4.1.4.

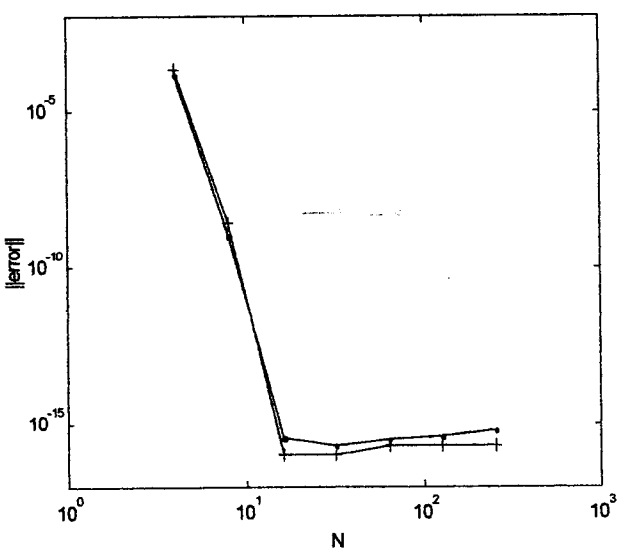

Figure 4.1.3 Error in the iterative solutions of (4.1) for $\mu = 1$.
Spectral integration (•), spectral differentiation (+).

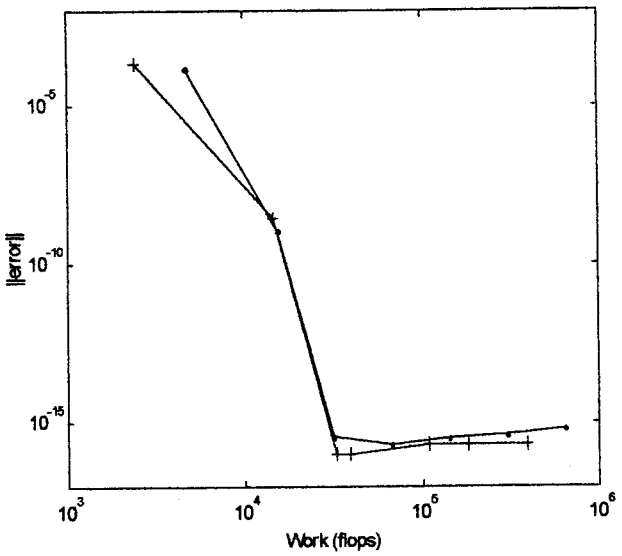

Figure 4.1.4 Efficiency of iterative solutions of (4.1) with $\mu = 1$. Spectral integration (•), spectral differentiation (+).

Table 4.2 Number of iterations.

N	SI iter.	SD iter.	κ_2
4	3	2	1.69
8	5	6	2.12
16	5	6	2.32
32	5	3	2.43
64	5	4	2.50
128	5	3	2.59
256	5	3	2.73

Next we consider the solution of (4.1) with $\mu = 100$, which is getting into the range of being a singular perturbation problem. The solution is nearly zero over most of the interval and possesses a sharp boundary layer at the right endpoint. For this problem the linear system obtained from the discretized integral equation is not nearly so well conditioned. The 2-norm condition number $\kappa_2(\boldsymbol{I} - 10^4 \boldsymbol{K}_N) > 4{,}000$. While this is not

large enough to cause serious difficulties with a direct solution, it does make iterative solution methods impractical. We found that the CGS, BCG, and GMRES iterative methods all failed to converge in under N iterations on this problem. We emphasize that for spectral integration to really be competitive with spectral differentiation in the solution of stiff boundary value problems a good preconditioner is needed, but has not yet been devised. We shall comment further on this issue in the next chapter.

In Figure 4.1.5 we compare the efficiency of the spectral integration and differentiation methods with direct solution of the linear system, and spectral differentiation with a preconditioned iterative solution. The solutions were computed with N = 16,32,64,128, and 256. The error vs. work curves are marked as in the previous example, with the exception of the iterative solution. For this problem the differentiation approach is significantly more efficient.

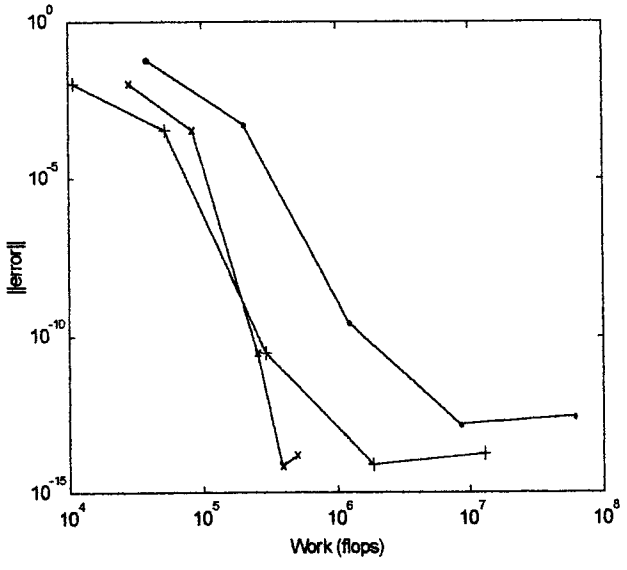

Figure 4.1.5 Efficiency comparison: solution of (4.1) with μ = 100.
Spectral integration (•), spectral differentiation direct(+), and iterative (x).

Note that the spectral integration solution with $N = 128$ requires roughly fifteen times as much work as the iterative spectral differentiation solution with the same N. Also note that for this problem the error with the integral approach is consistently larger for a given N than with the differentiation approach. This difference is presumed to be a consequence of the ill-conditioning of the matrix $\boldsymbol{I} - 10^4 \boldsymbol{K}_N$.

The CGS iterations were terminated when the 2-norm of the residual was reduced to less than 10^{-16}. The number of iterations executed for each N are listed in Table 4.3. A FFT based routine was used to apply the spectral differentiation operator and banded Gaussian elimination to apply the preconditioner.

Table 4.3 Number of iterations.

N	16	32	64	128	256
iterations	3	5	7	10	7

4.2 A variable-coefficient problem.

Our second example is a numerical solution of a variable-coefficient differential equation with a highly oscillatory forcing term

$$u'' - xu = f(x), \qquad u(-1) = 1, \quad u(1) = 2 \tag{4.2}$$

$$f(x) = 200\pi(1 - 3x^2)\cos(200\pi x) - \tfrac{1}{2}[6x + ((200\pi)^2 + x)(x - x^3)]\sin(200\pi x).$$

Where $f(x)$ has been chosen such that

$$u(x) = c_1 Ai(x) + c_2 Bi(x) + \tfrac{1}{2}(x - x^3)\sin(200\pi x)$$

$$\text{with } c_1 = \frac{2Bi(-1) - Bi(1)}{Ai(1)Bi(-1) - Ai(-1)Bi(1)}, \quad \text{and} \quad c_2 = \frac{Ai(1) - 2Ai(-1)}{Ai(1)Bi(-1) - Ai(-1)Bi(1)}.$$

Here Ai and Bi denote the two linearly independent solutions of the homogeneous

equation, the Airy functions of the first and second kind [1]. We use a built-in MATLAB function to evaluate the Airy functions.

The solution which oscillates two hundred times in the interval [-1,1] requires $N > 600$ for the spectral methods to even begin to resolve it. The large values of N required for an accurate solution of this problem make direct solution of the linear systems prohibitively expensive. The solutions are carried out with $N = 256, 512, 1024,$ and 2048 and serve to illustrate the impressive performance which can be achieved by coupling iterative methods with matrix-free transform based routines to apply the discrete integral and differential operators. Referring to Figure 4.2.1, we note that the solutions with $N = 2048$ require less work than a direct solution with $N = 256$ (refer to Figure 4.1.2). We observe that the accuracy achieved by the two approaches for given N is quite comparable. Note also that the spectral integration solution is obtained with considerably less effort.

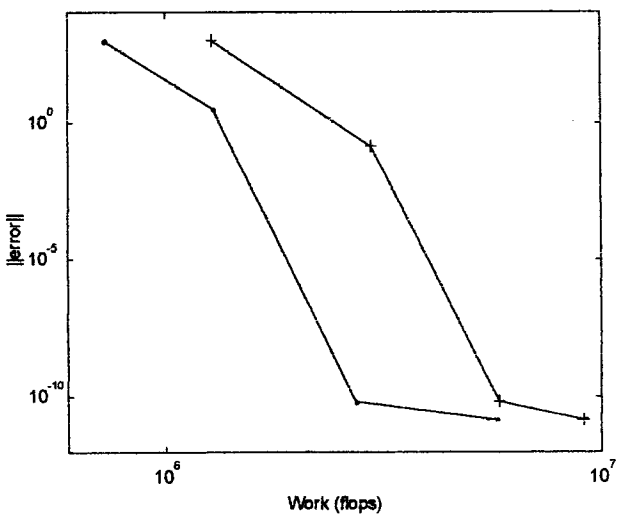

Figure 4.2.1 Error vs. work for spectral solutions of (4.2).
Spectral integration (•), spectral differentiation (+).

There are at least two factors which contribute to the larger amount of computational work of the spectral differentiation solution. First is the number of iterations required, which is a function of the condition number of the system. For the system of equations in the spectral integration method the CGS iteration converged in five iterations for all values of N. In the spectral differentiation case the preconditioned CGS iteration took at least eleven iterations in each case. A second reason for the increased work for the spectral differentiation solution is the extra work to apply the preconditioner at each iteration. Figure 4.2.2 shows a typical plot of the norm of the residual during the course of the CGS iteration for the spectral integration solution.

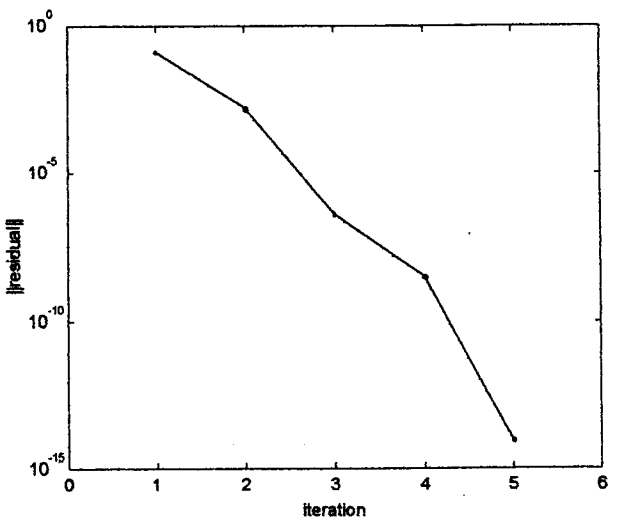

Figure 4.2.2 Convergence of the CGS iteration. Spectral integration solution of (4.2): $N = 2048$ (typical).

In regards to the question "Can spectrally accurate solutions to variable-coefficient boundary value problems be obtained in order $N\log(N)$ operations?" we offer Figure 4.2.3 as convincing evidence that the answer is yes, at least in this particular case. We note, however, that the implied constant may be quite large.

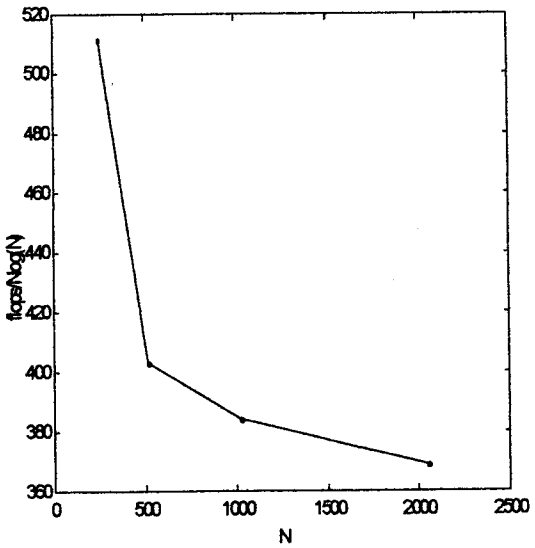

Figure 4.2.3 Work/Nlog(N) for the spectral integration solution of (4.2).

There is another advantage of the spectral integration approach and that is speed. Execution times are not customarily reported in the numerical analysis field, due to the fact that they are machine dependent. However the differences here are significant and so we break tradition by reporting them. The computations for problem 4.2 were done on a personal computer with a 100 MHz processor running un-compiled MATLAB 5.0. The execution times in seconds are reported in Table 4.4, the headings SI and SD refer to spectral integration and differentiation respectively.

Table 4.4 Execution times.

N	SI	SD
256	1.27	10.60
512	1.53	23.12
1024	1.92	48.86
2048	3.57	79.00

We attribute the large difference in execution times to the presence of seemingly unavoidable loops in the spectral differentiation codes. There are loops in both the spectral differentiation routine and the banded Gaussian elimination routine which applies the preconditioner.

4.3 Nonlinear boundary value problems.

In this section we present results from the numerical solutions of two nonlinear boundary value problems which admit closed form solutions. For our first example we consider the following equation

$$u'' = u'u, \qquad u(-1) = 0, \qquad u(1) = 2, \tag{4.3}$$

$$u(x) = c\tan(c(x+1)/2).$$

The constant c satisfies $c\tan(c) = 2$. This value rounded to fourteen places is 1.076 873 986 311 80. The spectral integration approach leads to the integral equation $u = x + 1 - J(u^2/2)$, and the discrete analogue

$$\boldsymbol{u} = \boldsymbol{x} + \boldsymbol{i} - \boldsymbol{J}_N(\boldsymbol{u} \odot \boldsymbol{u})/2. \tag{4.4}$$

Here $\boldsymbol{i} = (1,1,...,1)^T$, and $\odot$ denotes the Hadamard (or component-wise) product. We employ a simple fixed point iteration to solve (4.4). We note that a number of strategies can be employed to accelerate the convergence of this iteration, but from a programming standpoint the simplicity of fixed point iteration is unsurpassed. Starting from an initial vector which interpolates the boundary conditions $\boldsymbol{u}^0 = \boldsymbol{x} + \boldsymbol{i}$ we iterate; $\boldsymbol{u}^{n+1} = \boldsymbol{x} + \boldsymbol{i} - \boldsymbol{J}_N(\boldsymbol{u}^n \odot \boldsymbol{u}^n)/2$ for $n = 0,1,2,...,$ until the norm of the residual $\|\boldsymbol{u}^{n+1} - \boldsymbol{u}^n\|$ is acceptable. The iterations were terminated when $\|\boldsymbol{u}^{n+1} - \boldsymbol{u}^n\| < 10^{-15}$,

or when $n > N$ whichever occurred first. We note that for n up to $N-1$ this iteration emulates, in fact automates, the method of successive approximations or Picard iteration for approximating the solution of integral equations [13].

The spectral differentiation approach to this problem leads to the discrete equation $\boldsymbol{u} = (\overline{\boldsymbol{D}}^{(2)})^{-1}(\boldsymbol{u} \odot \boldsymbol{Du}) + \boldsymbol{x} + \boldsymbol{i}$, which is solved by fixed point iteration with starting vector $\boldsymbol{x}+\boldsymbol{i}$. Note that in this approach one essentially must solve a linear system at each iteration. An efficient way to implement this computation is to perform an LU factorization of $\overline{\boldsymbol{D}}^{(2)}$ at the start. Then at each iteration solve the triangular systems $\boldsymbol{Ly} = \boldsymbol{u} \odot \boldsymbol{Du}$, $\boldsymbol{Uu} = \boldsymbol{y}$ and then assign $\boldsymbol{u} \leftarrow \boldsymbol{u}+\boldsymbol{x}+\boldsymbol{i}$. We employ the same stopping criteria as for the spectral integration approach. Figure 4.3.1 shows the results of the iterations and Figure 4.3.2 gives details on how the iteration converges for the case $N = 32$. We point out that the spectral integration approach seems to be preferable for problems of this form because of the fact that there is no need to solve a linear system at each iteration.

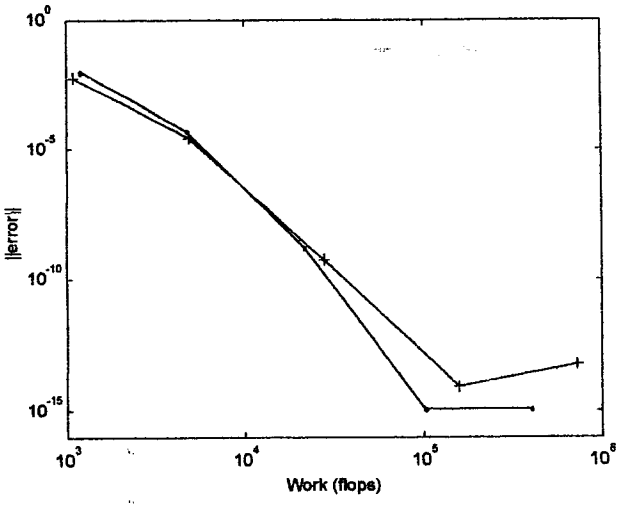

Figure 4.3.1: Solution of (4.3). Error vs.work for $N = 4, 8, \ldots, 64$.
Spectral integration (•), spectral differentiation (+).

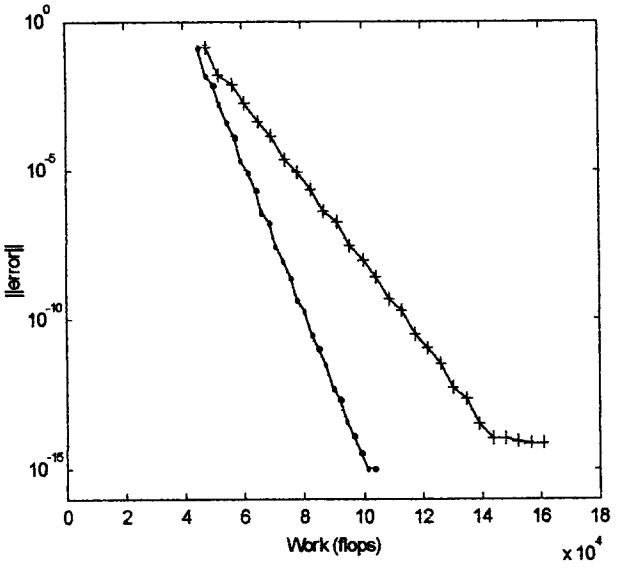

Figure 4.3.2 Progress of the iteration for $N = 32$.
Line markers indicate the status at successive iterations.
Spectral integration (•), spectral differentiation (+).

The second example of this section is the nonlinear boundary value problem

$$u'' = \exp(u) \qquad u(-1) = u(1) = 0, \tag{4.5}$$

$$u(x) = \log(2c^2 \sec^2(cx)).$$

Where the constant c is the solution of $c = \cos(c)/\sqrt{2}$. The value of c rounded to fourteen digits is 0.588 250 969 950 92.

The spectral integration approach leads to the integral equation $u = K\exp(u)$, and the analogous discrete equation $\boldsymbol{u} = \boldsymbol{K}_N \exp(\boldsymbol{u})$, where the exponential is taken to mean $(\exp(u_0),\ldots,\exp(u_N))^T$. The discrete equation is subjected to fixed point iteration with the same stopping criteria as in the previous example.. A zero vector is used to start the iteration and a transform based routine is used to apply the $\boldsymbol{K}_N$ operator.

In the spectral differentiation approach we obtain the discrete equation $\bar{D}^{(2)}u = exp(u)$, which is solved by fixed point iteration on $u = \left(\bar{D}^{(2)}\right)^{-1} \exp(u)$ implemented with an initial LU factorization as in the previous example. Refer to Figures 4.3.3 and 4.3.4 for the statistics on these computations.

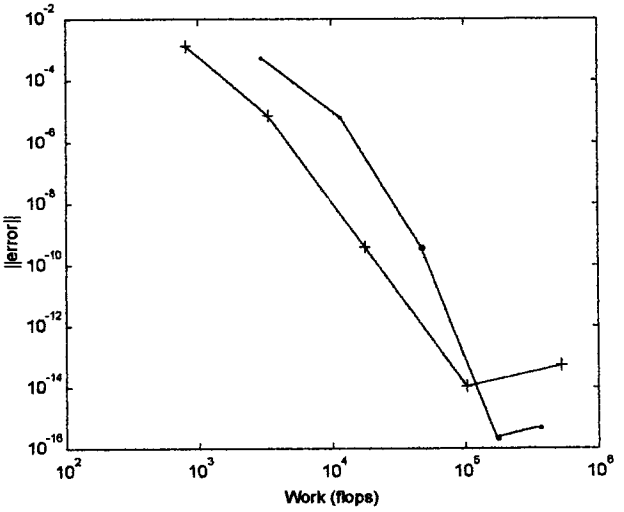

Figure 4.3.3: Solution of (4.4). Error versus work for $N = 4, 8, 16, 32, 64$. Spectral integration (•), spectral differentiation (+).

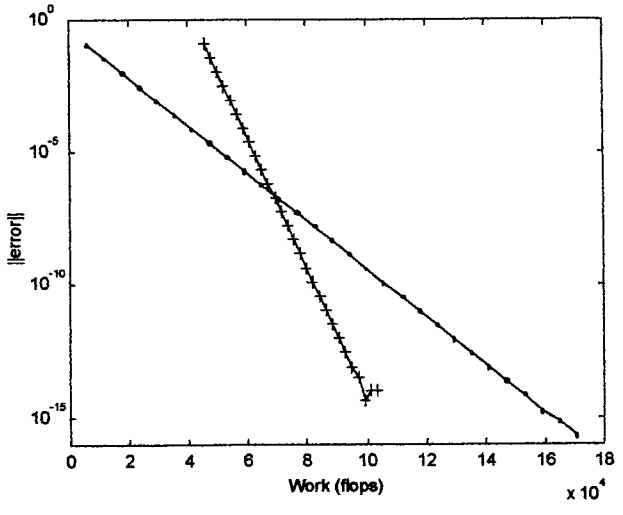

Figure 4.3.4 Progress of the iteration for $N = 32$. Spectral integration (•), spectral differentiation (+).

There are alternate methods available for the solution of the linear systems that arise in the spectral differentiation approach to these nonlinear problems. One alternative is to initially compute the inverse of $\boldsymbol{D}_N^{(2)}$ and then to simply multiply by $\left(\boldsymbol{D}_N^{(2)}\right)^{-1}$ at each iteration. A second alternative is to use a preconditioned iterative method as in problems (4.1) and (4.2). For N in the range used for problems (4.3) and (4.5) we have found that the approach of initially computing an LU factorization to be the most efficient of these three alternatives.

Based on the results of the test problems of this chapter we conclude that the spectral integration approach is a viable and powerful numerical method for the solution of two-point boundary value problems. We have found that in some cases spectral integration is very competitive with the more conventional spectral differentiation methods. These two methods seem to be complementary in the sense that neither one is clearly superior for all types of problems. However for problems with large coefficients it appears that spectral integration methods cannot compete with preconditioned spectral differentiation methods, at least not until an effective preconditioner is devised for the $\boldsymbol{K}_N$ matrices.

Chapter 5
Conclusions

This thesis has made a number of contributions to the development of spectral integration methods for the solution of boundary value problems. A stable $O(N^2 \log(N))$ algorithm has been developed for generating the spectral integration matrices $\boldsymbol{S}_l$ and $\boldsymbol{S}_r$ as well as the discrete integral operators $\boldsymbol{K}_N$ and $\boldsymbol{J}_N$. The theorem on the convergence of spectral integration presented in Chapter 2 has helped to provide a theoretical foundation for applications in the numerical solution of integral equations. We have proven a theorem on the spectrum of the discrete Chebyshev transform matrix which implies the stability of the transform and inverse transform. We have shown that spectral integration can be performed efficiently through the use of the FFT.

We have extended the applicability of the spectral integration approach to boundary value problems to include both Neumann and Robin type boundary conditions. The possibility of obtaining spectrally accurate solutions to variable-coefficient boundary value problems in $O(N \log N)$ operations has been demonstrated. Our comparison of the accuracy and efficiency of spectral differentiation and spectral integration is, to the best of our knowledge, the first to be done.

In the course of performing the computations for our comparisons it became apparent that for the integration approach to be truly competitive with spectral differentiation methods a good preconditioning scheme is needed. A good preconditioner must satisfy two requirements. First, the preconditioning matrix must be a reasonable approximation to the system being preconditioned. Secondly, the preconditioner must be easily inverted. In the case of finite-difference preconditioners for spectral differentiation operators these conditions are both satisfied. The difficulty in developing an analogous

scheme for spectral integration operators is that even low order quadrature formulas lead to dense matrix approximations to the integral operator K. Thus, it appears that a completely different strategy is called for in devising a preconditioner. We suggest developing an effective preconditioner for the $\boldsymbol{K}_N$ matrices as a challenging research project for anyone involved in the preconditioning field.

The $\boldsymbol{K}_N$ matrices possess many interesting structural properties (centro-symmetry being only one) which are not yet fully appreciated. A detailed theoretical study of these matrix properties could lead to a more complete understanding of spectral integration and to improved algorithms.

Some topics for possible future research in the spectral integration area include: applications to higher order boundary value problems and systems of boundary value problems, applications to boundary value problems for partial differential equations, and applications to integral equations in general. Applications to initial value problems for both ordinary and partial differential equations is also an area which has definite possibilities.

BIBLIOGRAPHY

1. Abramowitz, Milton,and Stegun, Irene A., eds. 1965. *Handbook of mathematical functions*. New York: Dover Publications.

2. Axelsson, Owe. 1996. *Iterative solution methods*. Cambridge: Cambridge University Press.

3. Canuto, Hussaini, Quateroni, and Zang. 1988. *Spectral methods in fluid dynamics*. Berlin Heidelberg: Springer-Verlag.

4. Fornberg, Bengt. 1996. *A practical guide to pseudospectral methods*. Cambridge: Cambridge University Press.

5. Funaro, Daniele. 1992. *Polynomial approximation of differential equations*. Berlin Heidelberg: Springer-Verlag.

6. Golub, Gene H, and Van Loan, Charles F. 1996. *Matrix computations*. Baltimore: Johns Hopkins University Press.

7. Greengard, L. 1991. *Spectral integration and two-point boundary value problems*. SIAM J. Numer. Anal. Vol. 28, No. 4: 1071-1080.

8. Kauthen, Jean-Paul.1998. *Solving Volterra equations is really that easy*! Proceedings of the HERMCA '98 conference, Athens Greece.

9. Reddy, Satish C. and Weideman, J.A.C. Oregon State University. *Accuracy of Chebyshev differencing for analytic functions*. manuscript in preparation.

10. Rivlin, Theodore J. 1990. *Chebyshev polynomials: from approximation theory to algebra and number theory*. New York: John Wiley & Sons.

11. Saad,Yousef. 1995. *Iterative methods for sparse linear systems*. Boston: PWS Publishing.

12. Tyrtshnikov, Evgenij E. 1994. *How bad are Hankel matrices*? Numer. Math. 67: 261-269.

13. Zwillinger, Daniel. 1989. *Handbook of differential equations*. New York: Academic Press.

APPENDICES

Appendix 1
The CGS Method

The linear systems which arise in spectral solutions to boundary value problems are typically nonsymmetric. Iterative methods for nonsymmetric systems fall roughly into three categories. First are methods based on the normal equations, for example the CGN method (conjugate gradient applied to the normal equations). Second are orthogonalization methods such as GMRES (generalized minimum residual method). A third class consists of biorthogonalization methods, such as the CGS (or conjugate gradient squared) method.

In many of the example problems considered we have used routines to compute matrix-vector products rather than explicitly creating the matrices. As the matrix transposes needed for CGN type iterations have not been readily available we restricted our consideration to the transpose-free CGS and GMRES methods.

A number of factors influenced our decision to employ the CGS method in our examples. One factor is that the CGS method has the desirable characteristic of requiring a fixed amount of work per iteration, while for GMRES the work per iteration increases with each iteration. For the linear systems which arose in the spectral integration approach we have observed that the required number of CGS iterations for a given problem is essentially independent of N, this fact coupled with the use of transform-based routines to perform matrix-vector products permits solutions in $O(N\log N)$ flops.

The final factor which influenced our choice was that for the set of problems considered CGS simply outperformed GMRES in terms of total work. We found that the truncation and restart scheme GMRES(k) could be competitive with the CGS iteration,

however we also discovered that the optimum restart parameter k was dependent on N. Faced with the added complexity of determining optimum (or near optimum) k for several different values of N for each problem, we decided to employ the parameter-free CGS method exclusively in our examples. For details on the CGN, GMRES, and CGS algorithms see [11] and the references therein.

The Conjugate Gradient Squared algorithm for solving $\boldsymbol{Ax} = \boldsymbol{b}$ [11]:

Compute $r_0 := b - Ax_0$; $\tilde{r}_0$ arbitrary.
$p_0 := u_0 := r_0$.
For $j = 0, 1, 2, \ldots$, until convergence:
$\alpha_j = (r_j, \tilde{r}_0)/(Ap_j, \tilde{r}_0)$
$q_j = u_j - \alpha_j A p_j$
$x_{j+1} = x_j + \alpha_j(u_j + q_j)$
$r_{j+1} = r_j - \alpha_j A(u_j + q_j)$
$\beta_j = (r_{j+1}, \tilde{r}_0)/(r_j, \tilde{r}_0)$
$u_{j+1} = r_{j+1} + \beta_j q_j$
$p_{j+1} = u_{j+1} + \beta_j(q_j + \beta_j p_j)$
End

In all computations we started the iterations with the initial vector $x_0 = b$, and set $\tilde{r}_0 = r_0$. In the computations for boundary value problem (4.1) the iterations were terminated when the residual norm $\|r_{j+1}\|_\infty$ was reduced below 10^{-16}, while for the variable coefficient problem (4.2) the stopping criteria was $\|r_{j+1}\|_\infty < 10^{-14}$.

Appendix 2
Spectral Differentiation Implementation Details

This appendix provides details on the implementation of spectral differentiation used in the examples of Chapter 4. MATLAB codes are provided for the transform-based spectral second derivative operator and the finite-difference preconditioner.

The code for spectral differentiation consists of three parts: a discrete Chebyshev transform, the application of the differential recurrence relation to the Chebyshev transform [4], and an inverse transform. The following MATLAB routine performs the spectral differentiation.

```
function d2u = chebdif2(u,N)
% FFT based spectral 2nd derivative
b = zeros(N+1,1); c = b;

% DCT
a = fft([u;u(N:-1:2)])/N;
a(1) = a(1)/2;a(N+1) = a(N+1)/2;

% apply recurrence relation to the transform
b(N) = 2*N*a(N+1);
b(N-1) = 2*(N-1)*a(N);
for i = N-2:-1:2
   b(i) = b(i+2)+2*i*a(i+1);
end
b(1) = b(3)/2+a(2);

% apply recurrence again
c(N-1) = 2*(N-1)*b(N);
c(N-2) = 2*(N-2)*b(N-1);
for i = N-3:-1:2
    c(i) = c(i+2)+2*i*b(i+1);
end
c(1) = c(3)/2+b(2);

% Inverse Transform
d = [c;c(N:-1:2)];
d(1) = d(1)*2;  d(N+1) = d(N+1)*2;
% remove imaginary part introduced by rounding errors
d2u = real(fft(d))/2;
d2u = d2u(1:N+1);
```

Preconditioning is a means of transforming a linear system $\boldsymbol{Ax} = \boldsymbol{b}$ into an equivalent system $\tilde{\boldsymbol{A}}\boldsymbol{x} = \tilde{\boldsymbol{b}}$ which is easier to solve by an iterative method. We use left preconditioning [11] where the original system is multiplied (on the left) by an approximate inverse of $\boldsymbol{A}$. The transformed system takes the form $\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b}$.

The use of finite-difference preconditioners for spectral methods is discussed in [3],[5]. The second order centered finite-differences are taken on the Chebyshev point grid. The entries of the second derivative matrix $\boldsymbol{R}$ are derived in [5]. To express $\boldsymbol{R}$ concisely it is convenient to first define the differences h and $\tilde{h}$

$$\begin{aligned} h_i &= x_{i+1} - x_i, \qquad i = 0,1,\ldots,N-1, \\ \tilde{h}_i &= (x_{i+2} - x_i)/2, \quad i = 0,1,\ldots,N-2. \end{aligned}$$

The non-zero entries of $\boldsymbol{R}$ are given by

$$\begin{aligned} R_{ii} &= \frac{-2}{h_i h_{i+1}}, \qquad i = 1,2,\ldots,N-1, \\ R_{i,i+1} &= \frac{1}{h_i \tilde{h}_{i-1}}, \qquad i = 1,2,\ldots,N-2, \\ R_{i+1,i} &= \frac{1}{h_{i+1} \tilde{h}_{i+1}}, \qquad i = 1,2,\ldots,N-2. \end{aligned}$$

A sparse storage format is employed with the non-zero elements of $\boldsymbol{R}$ stored by diagonals in vectors A, B, and C and accessed by row index

$$\boldsymbol{R} = \begin{bmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & C_{N-2} & \\ & & & A_{N-1} & B_{N-1} & \end{bmatrix}.$$

The following code generates the preconditioner $\boldsymbol{M} = \boldsymbol{R} - \mu^2 \boldsymbol{I}$ for problem (4.1).

```
global A B C
x = cos([0:N]*pi/N)';
h = x(2:N+1)-x(1:N);
ht = (x(3:N+1)-x(1:N-1))/2;

% Diagonals of M

% Subdiagonal
A = [0;1./(h(2:N-1).*ht(2:N-1))];

% Diagonal
B = -2./(h(1:N-1).*h(2:N))   -   (mu^2)*ones(N-1,1);

% Superdiagonal
C = 1./(h(2:N-1).*ht(1:N-2));
```

The following MATLAB routine applies the preconditioner and is based on a banded LU factorization [2].

```
function x = Minv(b)
%  Solves the tridiagonal system Mx = b.

global A B C
% allocate memory
n = length(b);
d = zeros(n,1);
x = d;y = d;

% Forward elimination: solve Ly = b
d(1) = B(1);
y(1) = b(1)/d(1);
for i = 2:n
   d(i) = B(i)-(A(i)*C(i-1))/d(i-1);
   y(i) = (b(i)-A(i)*y(i-1))/d(i);
end

% Back substitution: solve Ux = y
x(n) = y(n);
for i = n-1:-1:1
   x(i) = y(i)-C(i)*x(i+1)/d(i);
end
```

Appendix 3
Codes for problem (4.1)

The spectral integration and spectral differentiation codes both call the following CGS routine for solving linear systems.

```
function x = CGS1(b,tolerance)
% CGS algorithm utilizing routine "matvec" to compute matrix-vector products.
x = b;
r = b-matvec(x);
ros = r; p = r; u = r;
rtros = r'*ros;
while norm(r,inf) > tolerance
    w = matvec(p);
    alpha = rtros/(w'*ros);
    q = u-alpha*w;
    x = x+alpha*(u+q);
    r = r-alpha*matvec(u+q);
    beta = (r'*ros)/rtros;
    rtros = beta*rtros;
    u = r+beta*q;
    p = u+beta*(q+beta*p);
end
```

Routine SI4_1 computes the spectral integration solution of problem (4.1) for specified μ, α, β, and N.

```
function u = SI4_1(mu,alpha,beta,N)
%  Spectral integration solution of  u" - (mu^2) u = 0,  u(-1) = alpha,  u(1) = beta

global mu
x = cos([0:N]*pi/N)';

f = x *(beta-alpha)/2 + ones(N+1,1)*(alpha+beta)/2;      % Right hand side
u = CGS1(f , 1e-16);      % Solve the linear system
```

Routine matvec performs the matrix-vector products in the CGS algorithm for the spectral integration solutions.

```
function y = matvec(u)
global mu
y = u - (mu^2) * K_N(u);
```

Routine K_N applies the discrete integral operator K_N.

```
function Ku = K_N(u)
%  Routine to apply K_N
N = length(u)-1;
x = cos([0:N]*pi/N)'; one = ones(N+1,1);
b = zeros(N+1,1); % allocate memory

%  form complex integrand vector
v = (x + one) .* u + i * (x - one) .* u;

%  DCT
a = fft([v;v(N:-1:2)])/N;
a(1) = a(1)/2;  a(N+1) = a(N+1)/2;

%  Antidifferentiate
b(1) = a(2)/4;
b(2) = a(1) - a(3)/2;
b(3:N-1) = (a(2:N-2) - a(4:N))./[4:2:2*N-4]';
b(N) = a(N-1)/(2*N-2) - a(N+1)/(N*N-1);
b(N+1) = a(N)/(2*N);

%  Inverse  DCT
c = [b;b(N:-1:2)];
c(1) = c(1)*2;  c(N+1) = c(N+1)*2;
v = fft(c)/2;
v = v(1:N+1);

% Separate real & imaginary parts
v1 = real(v);  v2 = imag(v);

%  Apply projection
v1 = v1 - one * v1(N+1);
v2 = one * v2(1) - v2;

% Physical space products
Ku = ((x - one) .* v1 + (x + one) .* v2) / 2;
```

Routine SD4_1 generates the spectral differentiation solution of problem (4.1) for specified μ, α, β, and N. The routine Minv is called upon to apply the preconditioner. A program listing for Minv appears on page 56 of Appendix 2.

```
function  u = SD4_1(mu,alpha,beta,N)
% Spectral differentiation solution of u"- (mu^2)u = 0, u(−1) = alpha,  u(1) = beta

global A B C  mu
x = cos([0:N]*pi/N)';

%  Set up the diagonals of the preconditioner
h = x(2:N+1) - x(1:N);
ht = (x(3:N+1) - x(1:N-1))/2;
%  Subdiagonal
A = [0;1./(h(2:N-1).*ht(2:N-1))];
%  Diagonal
B = -2./(h(1:N-1).*h(2:N))   -  (mu^2)*ones(N-1,1);
%  Superdiagonal
C=1./(h(2:N-1).*ht(1:N-2));

%  Right hand side
f = x*(beta-alpha)/2 + ones(N+1,1)*(alpha+beta)/2;

% Solve the linear system
u = CGS1(Minv(f(2:N)),1e-16);
u=[0;u;0]+f;
```

The following routine performs the matrix-vector products in the CGS algorithm for the spectral differentiation solutions. It calls the spectral differentiation routine chebdif2 and preconditioner routine Minv. Program listings for these two routines are given in Appendix 2.

```
function y = matvec(u)
global  mu
N = length(u)+1;
u = [0;u;0];
z = chebdif2(u)  - (mu^2)*u;
y = Minv(z(2:N));
```